

Tietokonekulttuurin erikoislehti

**Superkoneita ja roskiskoneita**

**Internetin Isoveljeä pakoon**

**Suomalaiset tietokoneet  
Eskosta Elliin**



**Matka  
unohdettuun  
tulevaisuuteen**

- 3 Pääkirjoitus**
- 4 Batmud: verkkopelien veteraani**  
24-vuotias suomalainen seikkailu tenhoa yhä.
- 7 Kolumni – Tapio Berschewsky**  
Taas se puhuu niistä dindendoistaan.
- 8 Rautaromu resurssina**  
Uusi elämä eilispäivän elektroniikalle.
- 12 Välilyönnit ja vänkyrät**  
Esitämme yhden ratkaisun koodin muotoilun ikuisuusongelmaan.
- 15 Exactumin kasvihuone**  
Oppilaitoksen katolla kasvatettiin ruohoa.
- 16 Kuinka isoveli eksytetään**  
Perustason ninjakurssi opettaa astumaan verkossa kevyin askelin.
- 20 Parempi binäärihaku**  
Unohda puhelinluettelot – pomppimalla löydät varmemmin.
- 22 Pureudu audiovisuaaliseen dataan**  
Monialustainen multimediakieli ojentaa tahtipuikon ohjelmoijalle.
- 26 Suomitietotekniikan historia**  
Kotimaiset koneet vuosikymmenten varrelta.
- 31 Kolumni – Ville Oksanen**  
Tänä keväänä Suomella on tilaisuus edistyä tietoyhteiskuntana.
- 32 Visuaalinen simulaatio**  
Komeassa kopissa on kivempi kaahata.
- 38 Nopeimmat tietokoneet**  
Supertietokoneet jättävät jälkensä maailmaan.
- 44 Kirjaesittely – Bit Bang**  
Suomalaisen bittinikkarin tarina painettiin paperille.
- 46 Blender, osa 1**  
Avoimeen 3D-mallinnusohjelmaan tutustuminen alkaa taustojen kartoituksella.
- 48 OpenGL-ohjelmointi, osa 4**  
Oman grafiikkamoottorin rakentaminen.
- 51 Moniväristä kuusnelostaidetta**  
Commodoren perinteisin grafiikkatila kiinnostaa pikselinnyplääjiä edelleen.
- 52 Ritmanin Batman ja muut lepakkopelit**  
Skrolli levittää nahkasiipensä ja pujahtaa bittibättisten maailmaan.
- 57 Pelikatsaus – Aikaetsivä**  
Tapanimäen ensimmäinen peli on tarunhohtoinen. Onko se myös hyvä?
- 58 Esittely – Vampire 600**  
Kiihdytinkortilla alitehoinen Amiga lentoon.
- 60 Ei näin!**  
Pelikonsoli on tietokone. Tietokone on siis hyvä pelikonsoli.
- 62 Tulevaisuus, joka luvattiin**  
Missä ovat lentävä autoni ja robottihovimestarini?
- 67 Sarjakuva**





Ville-Matias Heikkilä  
päätoimittaja

## Kadonnutta ruohonjuuribittiä etsimässä

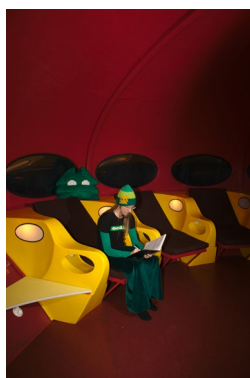
*Tietotekniikka monimutkaistuu jatkuvasti ja samalla vierauttaa todellisuudesta yhä pahemmin. Mikä neuvoksi?*

**M**oni pitkään tietotekniikkaa harrastanut on huomannut, että tekniikasta on yhä vaikeampaa saada otetta. Siinä missä takavuosien koneet suorastaan houkutelivat tutkimaan salojaan, nykyiset pikemminkin varjelevat käyttäjiään biteiltä ja nippeleiltä. Niiden esiin kaivaminen kaiken monimutkaisuuden seasta on yhä vaikeampaa ja turhauttavampaa. Ylpeimmätkin asiantuntijat nöyrtyvät lopulta siihen, etteivät käsitä suurinta osaa vempaintensa toiminnasta sen paremmin kuin taviksetkaan. Tekniikka vierauttaa omasta olemuksestaan pahemmin kuin koskaan.

Amerikkalaisfilosofi Albert Borgmann on kirjoittanut tekniikan vierauttavuudesta jo kolmisenkymmentä vuotta. Nykyihmisen elämä on täynnä rajapintoja, joiden takana piileksivän todellisuuden saa vapaasti unohtaa: ruoka tulee kaupan hyllyltä, vesi hanasta ja sähkö töpselistä. Vastalääkkeeksi Borgmann ehdottaa maanläheisiä askareita, joita hän kutsuu fokaalisiksi asioiksi ja käytännöiksi: puutarhanhoitoa, puulämmitystä, yhdessä syömistä. Askareita, joissa todellisuus kohdataan suoraan ja merkityksellisesti eikä minkään moduulirakenteen kautta.

Suomalaisille mökki on usein paikka, jonne mennään tekemään fokaalisia asioita ja unohtamaan arkimaailman abstraktiot. Arkimaailma karkaa kuitenkin yhä kauemmaksi mökkimaailmasta, eikä toinen enää auta hahmottamaan toista. Halkojen hakkaaminen ei paranna suhdetta aineettomaan automaatioon. Siksi tarvitaan myös toisenlaista ruohonjuuritason toimintaa – sellaista, joka ammentaa vierauttavana pidetystä tekniikasta itsestään eikä pyri sitä pakoon.

Mitä datakeskeisemmäksi tavallinen elämä muuttuu, sitä tärkeämmäksi tulee myös Skrollin edustama tietokonekulttuuri. Kotitarveohjelmointi, elektroniikkakorjailu ja tietokonetaidet eivät saa jäädä pelkiksi hakkerialakulttuurien salatieteiksi, vaan niitä on tuotava entistä rohkeammin esiin myös tavallisille ihmisille. Haitallisista stereotyyppioista on päästävä eroon, ja etenkin lasten ja nuorten on päästävä kunnolla sisään tietotekniikan olemukseen. Jokaisella on oikeus terveeseen bittisuhteeseen! 🌱



Kannen kuva:  
Mikko Rasa ja Mitol Berschewsky,  
Näyttelykeskus Weegee.



441 878  
Painotuote

### Skrolli

Tietokonekulttuurin erikoislehti

**Yhteydenotot** toimitus@skrolli.fi  
Ircnet: #skrolli

**Päätoimittaja** Ville-Matias Heikkilä  
**Toimituspäällikkö** Toni Kuokkanen  
**Toimitussihteeri** Ninnu Koskenalho  
**Taiteellinen johtaja** Risto Mäki-Petäys  
**Mediamyynti** Jari Jaanto  
**Talous** Anssi Kolehmainen

**Muu toimitus** Lauri Alanko, Mitol Berschewsky, Tapio Berschewsky, Mikko Heinonen, Jukka O. Kauppinen, Ronja Koistinen, Sade Kondelin, Teemu Likonen, Lauri Nurmi, Kalle Viiri

**Tämän numeron avustajat** Sampsa Alanen, Heikki Ketoharju, Simo Koivukoski, Jari Komppa, Antti Laaksonen, Albert Laine, Tapio Lehtimäki, Olli-Pekka Lehto, Jussi Määttä, Ville Oksanen, Juho Pietarinen, Annika Piironen, Manu Pärssinen, Ville Ranki, Mikko Rasa, Henna Ruohonen, Spiikki Sillanpää, Santeri Tani, Teija Tuhkio, Wallu

**Julkaisija** Skrolli ry

**Painopaikka** Tammerprint, Tampere,  
ISSN 2323-8992 (painettu)  
ISSN 2323-900X (verkkójulkaisu)



# Batmud: verkkopelien veteraani

*Batmud on tekstipohjainen verkkopeli, joka on pysynyt rakastettuna ajanviettotapana jo vuosikymmenien ajan. Mitä peliklassikolle on vuosien varrella tapahtunut?*

Teksti: Kalle Viiri

Kuvat: Annika Piironen, Mitol Berschewsky, Kalle Viiri

**M**ulti-User Dungeon, tutummin mud, on moninpelattava versio tekstiseikkailusta. Peligenre sai alkunsa Roy Trubshaw'n kehittämästä MUD-pelistä vuonna 1978. Pelin virtuaalimaailma, jossa pelaajat voivat temteltää vapaasti, innosti monia. Erilaisia mudeja alkoi pian syntyä, ja genrestä tuli

suosittu 1980-luvulla. Lopulta kehittyi koko nykyaikaisten verkkopelien genre.

Yksi vanhimmista ja yhäkin aktiivisesti toimivista mudeista on vuonna 1990 käynnistynyt suomalainen Batmud. Peliä on kehittänyt vuosien varrella yli 200 vapaaehtoista. Heidän työstään on kasvanut virtuaalimaailma, jonka laajuus ja monipuolisuus jättävät varjoonsa monet uudemmat verkkopelit.

## Batmudin vuosikymmenet

Batmud syntyi Teknillisen korkeakoulun opiskelijoiden harrastusprojektina, jota veti Janne Frösen. Virtuaaliseikkailut alkoivat huhtikuussa 1990, ja peli yhdistyi heti alkutaipaleellaan tamperelaisten teekkarien Herwood-mudiin. Virtuaalimaailmaa ylläpidettiin ja kehitettiin tässä vaiheessa noin 15 ohjelmoijan voimin.

Batmudin suosio levisi nopeasti, ja valtavan pelaajamäärän vuoksi pelin pyörittäminen yliopiston koneilla ei ollut enää mielekäästä. Niinpä pelin taustateksi perustettiin vuonna 1992 Balanced Alternative Techniques ry, joka kehittää ja ylläpitää Batmudia yhä tänäkin päivänä. Parempia palvelimia rahoitettiin

lahjoituksilla, mutta nekään eivät aina riittäneet. Liian suuren suosion vuoksi pelimaailmaan pääsyä piti usein rajoittaa.

Laitteisto on kehittynyt vuosien varrella huomattavasti. Alkujaan yliopiston koneella pyörinyt peli pitää nykyään majaa ammattilaistasoisella palvelimella. Sisältöä ja ominaisuuksia on vuosien varrella lisäilty. Peruspelin rinnalla pyörii muun muassa hardcore-versio, jossa pelaajahahmon kuolema on lopullinen.

Pelin kehittäjiin kuuluvan Niklas "Gore" Lindroosin sanoin "pelin kehityskaaresta kertoo jotain se, että se sai alkunsa jaetussa käytössä olleella Sony NWS-1530 -työasemalla, jossa oli 16 megaa muistia. Tällä hetkellä alla on Dell Poweredge 620 kahdella Xeon-prosessorilla ja 128 gigatavun muistilla." Lausahdukseen kiteytyy pelin pitkä historia.

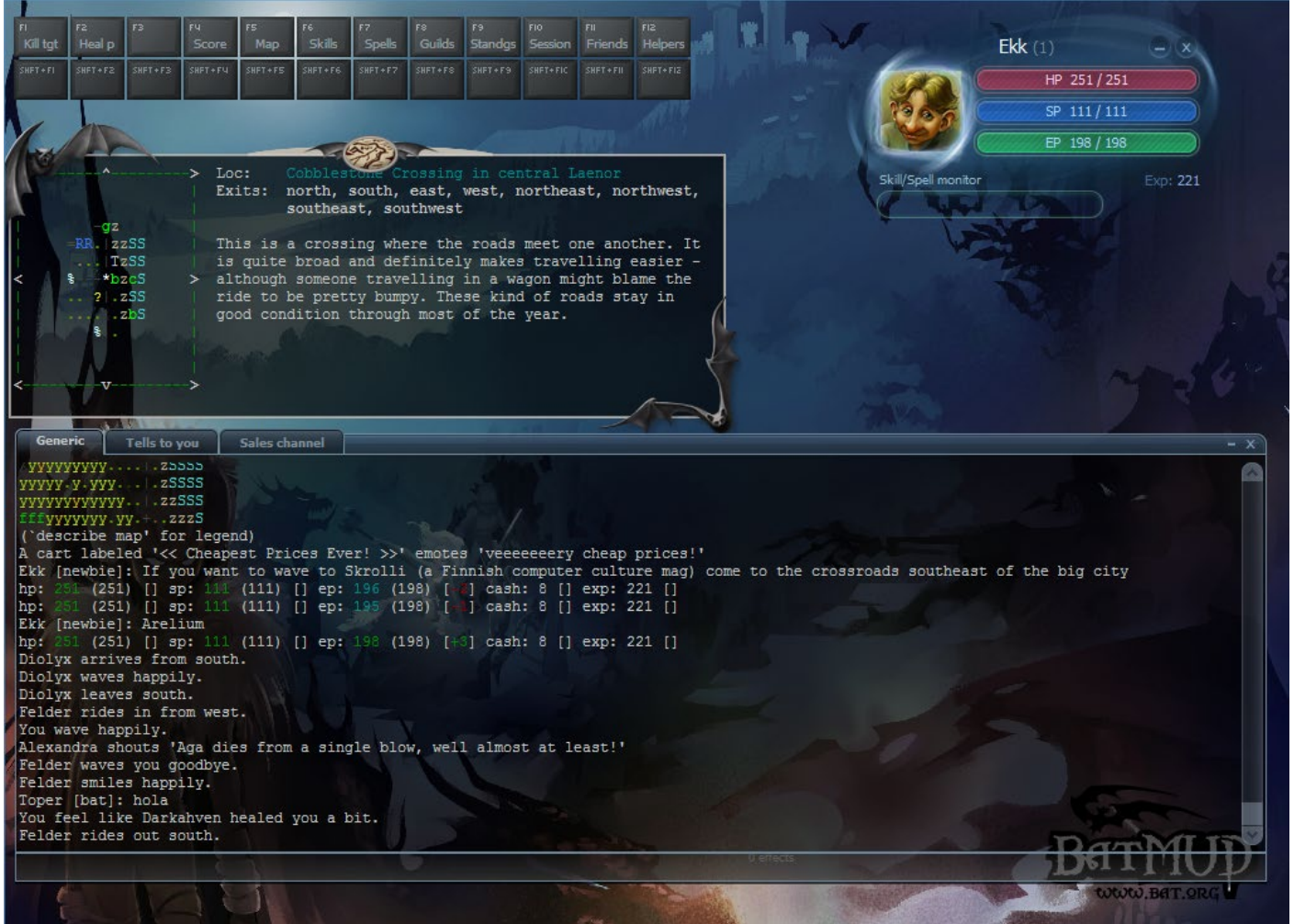
## Taustalla on väliä

Batmudin maailmassa riittää tekemistä. Pelin pääpaino on perinteisessä toimintaseikkailussa, ja erilaisia tutkittavia alueita hirviöineen ja aarteineen on sadoittain. Alueita on tarjolla jokaiselle

## Moninpeliä merkeillä

Mudit muistuttavat perustoiminnaltaan tavallista tekstiseikkailua. Pelaajia on yhden sijasta useita, ja kaikki toimivat reaaliaikaisesti samassa maailmassa. Pelaaja syöttää komennon – esimerkiksi "travel east" tai "attack dragon" – ja saa peliltä vastaukseksi tekstin, joka kertoo, miten maailma reagoi pelaajan toimiin. Vastauksessa on joskus hyvinkin tarkkaa tietoa maailman tilasta, esimerkiksi kuun vaihe ja säätila. Vastauksen ymmärtäminen vaatii jonkin verran harjoittelua ja kokemusta.

Vaikka graafiset verkkomoninpelit ovat nykyään valtavirtaa, toiminnassa on edelleen kymmenittäin roolipelaamiseen, toimintaseikkailuun tai pelaajien keskinäiseen kamppailuun keskittyviä mudeja.



kokemustasolle, helpoista vasta-alkajien lammastarhoista vaarallisiin linnoihin ja luolastoihin, joissa ei pärjää ilman korkeatasoisista seuruetta.

Roolipeleille tuttuun tapaan vihollisten surmaamisella tienaa kokemuspisteitä ja saalista. Kokemuspisteitä voi ansaita myös vieraillemalla uusissa paikoissa ensimmäistä kertaa. Pelaajan ei siis välttämättä kannata jäädä lypsämään kokemusta vain suosikkiluolastaan, vaan on hyödyllistä seikkailla myös mukavuusalueensa ulkopuolella.

Hahmon kyvyt taistelussa ja sen ulkopuolella määräytyvät rodun, taustan ja kiltavalintojen mukaan. Batmudissa on

epätyyppillisen paljon rotuja. Tuttujen örkkien, ihmisten ja haltioiden lisäksi luolastossa samoilevaan seurueeseen saattaa kuulua myös muumi, ankkat tai pingviini. Hahmon tausta voi olla hyvä uskonnollinen, paha uskonnollinen, paimentolainen, maaginen tai sivistynyt. Jokaisella näistä on omat piirteensä ja painotuksensa.

Tausta määrittää tärkeimpänä kuitenkin sen, mihin kiltoihin hahmo voi liittyä. Kilta muodostaa ison osan hahmon identiteetistä. Esimerkiksi hyvä uskonnollinen hahmo voi menestyä Tarmalen-parantajakultissa tai epäkuolleita tuhoavana nunnana. Paimentolaisesta voi tulla joko

raskaasti panssaroitu lihamuuri taikuri-toveriensa suojaksi tai villisti kirveellään vihollisia puolittava barbaari.

Kaikki hahmot eivät kunnostaudu taistelussa lainkaan. Sivistystaustaiset hahmot voivat vaikka ryhtyä kauppiaksi ja tienata korjaamalla toisten varusteita ja myymällä valmistamia tavaraita.

Killan päämajassa hahmo voi käyttää kokemustaan taitojensa ja taikojensa varmuuden ja tehon kehittämiseen. Killassa edetään kokemustasojen avulla. Tarjotut taidot vaihtelevat killan mukaan, mutta listoilta löytyy kaikkea aina tuhoisista erikoishyökkäyksistä rahan vaihtamiseen ja leiriytymiseen.

Taitojen ja taikojen lisäksi pelaajille tarjoillaan usein myös sivuhaasteita. Esimerkiksi barbaarien kiltta järjestää säännöllisin väliajoin verisiä taisteluita. Kiltta kannustaa jäseniään keräämään hurjaa soturimainetta aiheuttamalla mahdollisimman hirveää tuhoa ja kaaosta.

## Yhdessä kohti kunniaa

Hahmojen erikoistumisen takia ryhmätyö on iso osa Batmudin henkeä. Barbaarit, ritarit ja muut lähitaisteluhahmot pärjäävät melko hyvin heikompia vihollisia vastaan, mutta suurempia mörköjä vastaan pelaajien on muodostettava ryhmiä. Ryhmä liikkuu ja taistelee ryhmän perustajan tai hänen nimittämänsä johtajan mukana. Ryhmän muiden jäsenten sor-

```
Ekk hits Ugly magpie once making small marks.
hp: 220 (252) [-4] sp: 111 (111) [] ep: 197 (197) [] cash: 52 [] exp: 249 []
Ugly magpie is slightly hurt (70%).
***** Round 13 *****
Ugly magpie hits Ekk once causing a small scratch.
hp: 214 (252) [-6] sp: 111 (111) [] ep: 197 (197) [] cash: 52 [] exp: 249 []
Ugly magpie is slightly hurt (70%).
You are done with the chant.
You boom in sinister voice 'gtzt zur fehh'
You watch with self-pride as your magic missile hits Ugly magpie.
hp: 214 (252) [] sp: 95 (111) [-16] ep: 197 (197) [] cash: 52 [] exp: 249 []
***** Round 14 *****
Ugly magpie hits Ekk once making small marks.
Ekk hits Ugly magpie once causing a small scratch.
hp: 210 (252) [-4] sp: 95 (111) [] ep: 197 (197) [] cash: 52 [] exp: 249 []
Ugly magpie is in very bad shape (15%).
***** Round 15 *****
Your armours absorb some of the damage.
Ugly magpie hits Ekk once making small marks.
Ekk hits Ugly magpie once making small marks.
hp: 209 (252) [-1] sp: 95 (111) [] ep: 197 (197) [] cash: 52 [] exp: 249 []
Ugly magpie is in very bad shape (15%).
```



met jäävät vapaaksi taikojen ja taitojen käyttämiseen, kun itse taistelu tapahtuu automaattisesti.

Parhaimmillaan rytke voi olla hyvin intensiivistä. Niinpä jokainen ryhmän jäsen tarvitsee reilusti itsekuria, jotta ei livistä paikalta, kun vihollisen tulipallot vievät kestopisteet kriittisen punaiselle.

Pahimmassa tapauksessa kuolema yllättää. Tällöin tyyppillisesti joku pelimaailman parantajista tarjoutuu nostamaan vainajan kuolleista. Kuoleman seurauksena hahmo menettää ison osan käyttämättömistä kokemuspisteistään ja kärsii tilapäisestä heikkoudesta. Parantaja puolestaan saa palkkiokseen osan menetetyistä pisteistä.

### Vaihtelu virkistää

Jos väkivaltainen kuolema pelottaa, Batmudin maailmassa on paljon muutakin hauskaa ja erikoista koettavaa. Hirviöltä ansaittuja rahoja voi esimerkiksi tuhlatella kasinolla. Edistyneempi pelaaja voi hankkia oman laivan tai vaikka kokonaisen kaupungin. Välillä Batmudin arkea piristävät myös satunnaistapahtumat kuten barbaarien hyökkäykset tai meteorimyrskyt.

Huumorikin kukoistaa. Viholliset huutavat vitsikkäitä pilkkoja surmatesaan pelaajia. Tarkkasilmäinen havaitsee siellä täällä ovelia populaarikulttuuriviitauksia. Yksi pelin alueista on esimerkiksi teemoitettu Monty Pythonin Holy Grail-elokuvan mukaiseksi. Pelin suomalaiset juuret tulevat esille suomenkielisistä loitsuista, joita hahmo lukee taikoessaan, sekä tietenkin pelimaailmaa täplittäivistä saunoista.

Jos pelailu nykyisellä hahmolla kylästyttää, voi hahmonsa luoda uudelleen,

niin että rotu ja tausta vaihtuvat. Tällöin peli alkaa alusta, mutta pelaaja säilyttää suurimman osan saamastaan kokemuksesta ja saa loppujakin takaisin hiljalleen. Pelin voi siis kokea usealla eri tavalla ilman, että tarvitsee palata ihan kokonaan alkupisteeseen. Tämä reinkarnaatioksi kutsuttu ominaisuus lieventää hahmonluonnon tuskaa ja paineita siitä, ettei ensimmäinen hahmoluokka ehkä olekaan se kaikkein paras ja omaan pelityyliin sopivin.

### Yhteisössä on voimaa

Batmudin yhteisössä on noin 5 000 aktiivista pelaajaa, mikä nostaa sen suosituimpien mudien joukkoon. Vaikka suuri osa pelaajista on suomalaisia, on Batmud-yhteisö aktiivinen myös esimerkiksi Yhdysvalloissa, Ruotsissa ja Iossa-Britanniassa. Aktiivisen yhteisön kesken seikkailaan pelimaailmassa ja rupertellaan mukavia. Osa pelaajista käy Batmudissa lähes yksinomaan juttelemassa.

Välillä välit kuitenkin kiristyvät. Pelimaailman puitteissa ja kohtuuden rajoissa saa huijata, ryöstää toisten tavaroita ja tappa toisia pelaajahahmoja. Pelissä on nähty jopa suurten ryhmittymien välisiä sotia. Pelaajayhteisön yhteishenki on kuitenkin mainio. Batmudin pelaajat järjestävät usein tapaamisia tosielämässäkin, ja osa pelaajista on löytänyt peliyhteisön kautta töitä tai jopa puolison.

Batmud ei ole helppo peli, mutta se yrittää olla mahdollisimman ystävällinen aloittelijalle. Kokeneet pelaajat jakavat innokkaasti neuvoja yhteisön uusille jäsenille, ja pelin ohjetiedoista löytyy ratkaisu moneen kiperään kysymykseen. Mudille hieman epätyypillisesti Batmudissa on tekstikuvauksen lisäksi monelle

alueelle myös Ascii-merkeistä koostettu kartta, jonka avulla pelaajan on helpompaa suunnistaa paikasta toiseen.

### Vanha mestari, uudet haastajat

Kilpailu verkkopeleissä on kovaa, mutta Batmud pitää pintansa. Tekstikäyttöliittymä yhä porskuttaa uudempien graafisten kilpailijoidensa rinnalla. Sitä tukee myös virallinen selaimessa toimiva pääteohjelmisto Batclient, joka lisää käyttöliittymään modernia näppäryyttä. Se mahdollistaa esimerkiksi viestien jakamisen erillisiin ikkunoihin sekä lisää tuen erilaisille pelaamista helpottaville makroille ja skripteille.

Pelin kehittäjäryhmään pitkään kuulunut Niklas Lindroos muistuttaa tekstipohjaisuuden voimasta: yksikin koodari voi toteuttaa hyvän ideansa, siinä missä modernimpi verkkoroolipeli tarvitsee kokonaisen tiimin toteuttamaan uutta sisältöä. Tämän vuoksi Batmud pysyy jatkossakin yhtenä verkon runsassisältöisimmistä peleistä. 🏰

Skrolli kiittää:

Niklas "Gore" Lindroos.

### Batmudin historian erityisiä hetkiä

**2007** - Pelimaailman "pääkaupunkiin" Arelumiin hyökkää valtava lauma vihamielisiä hirviöitä, barbaareja ja muita valloittajia. Pelaajat kutsutaan talkoovoimin puolustamaan kaupunkia, tappamaan vihollisia ja siivoamaan taisteluiden jälkiä.

**2006** - Batmudin pelimaailma huuhtoutuu mereen. Pelaajat pelastautuvat uuteen maailmaan, joka koostuu viidestä mantereesta. Jokaisella mantereella on oma teema, ja jokainen manner vastaa kokoluokaltaan vanhaa pelimaailmaa. Rikkaat pelaajat reissaavat mantereelta toiselle omilla laivoillaan muiden tyytyessä julkisiin lauttoihin.

**2002** - Pahan Burglefloogah-jumalan teurastusreissusta johtuva sisällissota jakaa pelaajat kahteen leiriin. Hyvät pelaajat yrittävät surmata jumalan, ja pahat pelaajat pyrkivät parhaansa mukaan suojelemaan tätä. Yli 13 tuntia jatkunut yhtäjaksoinen kamppailu pelaajien kesken ja Burglefloogahia vastaan päättyi lopulta hyvän voittoon. Häviön seurauksena pahojen pelaajahahmojen tait eivät toimi kunnolla useaan viikkoon.



## Ostin sälää, pelasin

Tapio Berschewsky

*Tämä ei ole tyypillinen Skrolli-juttu. Käsittelen nimittäin tiukasti suljettua kaupallista tuotetta, joka on saatavilla jokaisesta Prismasta.*

Olen arvostellut siihen yhteen tekniikkalehteen aika monta tuotetta. Satoja pelejä ja tekniikkaleluja on käynyt käsieni läpi, ja olen sanonut niistä sanottavani. Osalle olen antanut satoja tunteja, toiset on täytynyt käsitellä hyvin nopealla aikataululla. Tiedän tehneeni ajoittain erinomaista työtä, joskus taas kiirettä vasten hutiloinut. Niin kai kaikki.

Valtaosaa näistä tuotteista yhdistää se, etten ole maksanut niistä mitään. Ne on toimitettu työpöydälleni ilmaiseksi. Joitain poikkeuksia toki on. Esimerkiksi Nintendo DS:n New Super

Mario Brosia

odotin niin

kovaa, että pistin ystävän tuomaan sen jenkeistä viikkoa ennen kuin arvostelukappale ilmestyi toimitukseen. Nämä ovat harvassa.

Ilmaista lelua pursuavasta norsunluuornista huuteleminen ei ole ongelmatonta. Kun omalla lompakolla ei käydä, katoaa käsitys siitä, mikä on rahan arvoista. Koskaan ei koe kutkuttavaa pettymystä, kun tuote ei vastaa siihen

kulutettua rahamäärää. Työkiireessä ei myöskään pääse näkemään pitkällisen käytön mukana tulevia ongelmia.

### Amatöörit apuun

Ongelmaan on ehdotettu ratkaisuksi lukija-arvosteluja. Eräs kollegani naureskeli taannoin Facebookissa ajatukselle. Ihmiset kun eivät ole hyviä myöntämään, että tekivät tyvästi ja tuhlasivat paskaan.

Mitä tahansa tekniikkakeskustelua seuraamalla huomaa kyllä, että varmaa on vain mielipiteiden poteroituminen ja ääliömäinen asemasota. Urpot varmaan tappelevat

Androidin, iOS:n ja Windows Phonen

nokkimisjärjestyksestä vielä lukemattomia vuosia, vaikka ne ovat kaikki suurin piirtein yhtä huonoja.

Tosin kovinkaan moni tekniikkatuote ei edes ole markkinoilla tarpeeksi pitkään, jotta niiden arvioinnista olisi iloa siinä vaiheessa, kun käyttöä on takana mustan vyön verran. Hyvä kun käytettynä saa. Jännä poikkeus tähän on tuoteryhmä, jota monet tekniikk-

kaintoilijat pitävät ärsyttävänä turhakkeina: pelikonsolit.

### Sieniä, kukkia, tulta ja tähtiä

No, itsehän tykkään Nintendon peleistä. Toisia ne eivät kiinnosta, koska niissä ei ammuta realistisilla aseilla, kaahata kovaa kiiltävillä tosimaailman urheiluautoilla tai pyritä tappamaan harmaanruskeissa ympäristöissä suuria määriä virtuaalisia idiootteja. Minusta ne ovat loistavia juuri sen vuoksi. Niinpä pelaan Nintendon konsoleilla.

Ostin syksyllä Wii U:n. Oikeastaan siksi, että sille julkaistiin hd-remasteroitu versio vuoden 2001 klassikosta, Wind Wakerista. Tiedän että mielipiteeni on väärä, mutta minulle se on definiitivinen Link-kokemus.

Halusin takaisin seilamaan vailla päämäärää, kuuntelemaan yömusiikin muuttumista aamumusiikiksi ja edelleen päivämusiikiksi. Halusin karkuun pimenevää syksyä sellaisiin maastoihin, joiden värikylläisyydestä voi Suomen luonnossa nauttia vain hallusinogeeneilla. Siksi upotin yli kolmehunttia rahaa taas yhteen pelilaitteeseen, vaikka kämppä on niitä jo täynnä.

Viimeinen Triforcen pala on yhä hakematta, mutta eiköhän sekin päivä vielä tule. Vaan olipa maukkaat 30 tuntia. Sen jälkeen olen tahkonnut Wii U:lla Super Mario 3D Worldin sekä Assassin's Creed IV: Black Flagin.

### Tämä on tuotearvio-osuus

Konsoli itsessään on aika yhdenmukainen. Se ei ole kiinnostava, eikä sen ohjain hetkauta suuntaan eikä toiseen. Paitsi siinä mielessä, että menin ostamaan sen helvetin Wind Waker -bundlen, jossa kolhon ohjaimen reunaan on printattu siroja kultaisia symboleja. Kenenköhän mielestä tämä on nättiä.

Nyt Wii U on jo pölynyntymismoodissa suurimman osan ajasta, koska pelattava loppui kesken. Vaikka sille on uudemmissa konsoleista eniten sisältöä. Katsotaan sitten taas loppusyksystä, josko ostaisin läjän uusia Wii U -pelejä. Tai ehkä sen PS4:n.

Mutta kyllä kuulkaas oli nautinnollista vetää Mariossa turripuku päälle heti ekassa kentässä. Siinä kun on kissa-asu. Sitten vielä sanovat, ettei Nintendo osaa internetiä!

Tuotearvio on pohjimmitaan aika epäskrollimainen artikkelityyppi. Haluaisin silti nähdä sellaisia Skrollin nimen alla. En lehdessä, vaan saitilla. Enkä niitä pintaraapaisuja, mitä tekniikkalehdet yleensä julkaisevat.

Kaipaan teidän Skrollin lukijoiden omia arvosteluja tuotteista, joita olette käyttäneet pitkään, hartaasti ja kunnolla. Ei vain nopeasti kokeilemalla, vaan ottamalla osaksi elämää.

Pistäkää toimitukselle mailia, jos toivotte samaa. Palaute siirtää maailmaa. 🐘

# Vapauta rauta roskiksesta

*Käytöstä poistunut elektroniikka päätyy usein kasvattamaan jätevuoria. Kierrätyksen ja korjauksen kautta monet laitteet ja komponentit voisivat kuitenkin jatkaa elämäänsä.*

Teksti: Albert Laine Kuvat: Risto Mäki-Petäys, Andrew Gryf Paterson

Jokapäiväinen elämämme sujuu erilaisten välineiden avustamana. Heräämme yhden vempaimen avulla, keitämme toisella aamukahvimme ja ajamme kolmannella töihin. Laitteiden avulla saamme ja jaamme tietoa, vaikutteita ja ideoita.

Ihmislajilla on ilmiömäinen kyky rakentaa työkaluja ja tehostaa työtä. Tämän lajipiirteiden seurauksena on syntynyt ympäristö, jonka olemme itse muovanneet ja jota myös sivilisaatioksi kutsutaan. Seurausta on sekin, että elämäntapamme tuottaa aina vain lisää esineitä. Vanhentuneet ja vialliset yksilöt poistetaan käytöstä, vaikka osaavissa käsissä niiden tarinalle voisi löytyä jatkoa.

## Kuumailmapuhaltimen parantava voima

Olen hyödyntänyt noin vuosikymmenen ajan laitteita, jotka on heitetty tarpeettomina pois. Työasema, jolla kirjoitan tätä artikkelia, on dyykattu. Teinit olivat käyttäneet konetta maalikuulapysyjensä harjoituskohteena. Kone oli kiintolevyä vailla, mutta muut komponentit olivat tallessa. Kone on Dell Optiplex 760, jossa on neliytiminen Core 2 -prosessori, GTX 460 -näytönohjain sekä 8 GB DDR2-muistia. Ei ihan uusinta uutta, mutta hinta-laatu-suhteella ei ole voittanutta. Pikainen nettihaku kertoi, että kone oli vain kaksi vuotta vanha. Ei pöllömpää!

Ensimmäisellä käynnistyksellä kone boottasi, mutta se sammui muutaman

minuutin käytön jälkeen. Ongelma toistui myös toisella virtalaitteella. Asiaa pohdittuani tulin siihen tulokseen, että jokin muistimoduuleista oli varmastikin vioittunut.

Memtest-ohjelman ajettuani havaitsin, että toinen 2 GB:n DDR2-muistimoduuleista sekosi lämmentyään. Lueskelin netistä vinkkejä ongelman ratkaisemiseksi. Päätin soveltaa näytönohjainten ja pelikonsolien kotikorjaustapaa, "uunnittamista", jossa viallinen laite kuumennetaan pariinsataan asteeseen. Uunia

parempi ratkaisu on käyttää lämpötilanturia sekä foliota eristämään elektrolyyttikondensaattorit sekä muoviset osat, jotka eivät kestä 200 asteen lämpötilaa.

Päätin kokeilla kuumailmapuhallinta vialliseen muistimoduuliin. Asetin muistimoduulin kahden puupalan päälle ja aloin lämmittää sitä matalalla lämmöllä. Pikkuhiljaa nostin lämpöä kahdensadan asteen tavoitelämpötilaan. Yllättäen muistimoduuli naksautti kuuluvasti, ja arvelin sen olevan mennyt kalua. Päätin kuitenkin kokeilla moduulia eräässä



Cindy Kohtala Sortti-aseman tietokonejätösten äärellä.



emolevyssä. Annoin Memtestin rassata muistimoduulia muutaman tunnin ja huomasi ilokseni, että se toimi virheettään. Siirsin moduulin takaisin löytämäni koneeseen - ja samainen muisti on vieläkin käytössä!

Tässä tapauksessa kyse oli selvästä valmistusvirheestä. Samalla tavoin voittunutta muistimoduulia en ole sittemmin kohdannut. Internet tarjosi suunnan ratkaisulle, mutta suoraa esimerkkiä RAM-moduulien elvyttämiseen kuuma-ilmapuhaltimella en löytänyt.

Muistimoduulin onnistunut korjaaminen sai minut kokeilemaan kuuma-puhallinmenetelmää muihinkin löytämiini komponentteihin. Tekniikka toimii näyttöohjaimiin, joiden kuvassa näkyy shakkilauta-artefakteja tai kuva puuttuu tyystin. Samalla tekniikalla olen saanut lukuisia kannettavien tietokoneiden emolevyjä käyttökuntoon. Monet laskenta-käytössäkin tehokkaat GTX/GTS 8800 -näyttöohjaimet kärsivät samasta vaivasta.

Olen kasannut tällä tavoin korjaamistani dyykkikomponenteista kaikkiaan 13 Dual Core -konetta, jotka on varustettu 500-1000 gigaflopin laskentatehoisilla grafiikkasuorittimilla. Vaikka töiden määräaika olisi vaarallisen lähellä, työhuoneeni laskentafarmiin voi luottaa: se rouskuttaa kuvat valmiiksi aina ajallaan.

Pelikonsoleissa, kannettavissa tietokoneissa ja näyttöohjaimissa käytetään usein BGA-pintaliitoskomponentteja, jotka kokemukseni mukaan voittuvat Euroopan unionin RoHS-direktiivin vuoksi. Säästösten takia juotoksissa siirryttiin vuoden 2006 tienoilla käyttämään lyijytöntä tinaa, mutta valitettavasti tehokkaiden komponenttien voimakas lämpötilavaihtelu murtaa juotoksia 2-3 vuoden käytön jälkeen - juuri sopivasti korvaavan tekniikan tullessa markkinoille.

Ympäristösäädos ampuikin lopulta äiti Maata jalkaan moninkertaistamalla elektroniikkajätteen määrän ja lisäämällä uusien laitteiden valmistusta ja myyntiä. Laitevalmistajat voittivat jälleen.

## Duopoliin valta

Duopoli on kapitalistisessa markkinatalousjärjestelmässä yleinen ilmiö. Se tarkoittaa tilannetta, jossa kaksi kilpailevaa yritystä tarjoaa samaa palvelua tai tuotetta. Käytännössä kaksi markkinajohtajaa kommunikoi keskenään jakaakseen markkinoita ja rytmittääkseen uusien tuotteiden julkistamisen. Valmistajat tarjoavat saman suorituskyvyn ratkaisuja hintaluokissaan, vaikka arkkitehtuurissa saattaa olla suuria eroja.



Hylättyjä mutta tehokkaita näyttöohjaimia. Kuvassa 8800 GTX, 8800 GTS, GT 240 sekä 9800 GT.

Teknologiset innovaatiot ja valmistusmenetelmät etenevät harppauksin. Esimerkiksi uudentyypin näyttöohjainpiirin suunnittelu kestää vuosia. Yrityksen on saatava tuloja myös kehitystyön aikana, joten myynnin on oltava tasaista. Näyttöohjainpiirien valmistajat Nvidia ja AMD haastettiin muutama vuosi sitten oikeuteen duopolimaisesta kartellitoiminnasta. Laittomasta kuluttajien rahojen lypsämisestä oli vahvaa näyttöä, ja syytetyt sovittelivat jutun miljoonakorvauksin. Koska tapauksesta ei tullut oikeuden päätöstä, sai toiminta kuitenkin jatku.

Kuluttajien muisti on lyhyt. Eilispäivän huipputeknologia löytää pian tiensä jätevuoriin, jotka ovat jo ennestään suuria. Yksi oiva esimerkki on h.264-videon laitteistopurku. Tällä ominaisuudella on myyty näyttöohjaimia vuodesta 2004

lähtien. Esimerkiksi Geforce 6600:ssa on kyseinen ominaisuus, mutta se toimii vain tietyllä ajuriversiolla ja tietyn video-ohjelmien tietyllä versiolla. Kun asiakas päivitti ajurit, toistosoftan tai käyttöjärjestelmän uudempaan, ominaisuus lakkasi toimimasta. Kone siis alkoi tökkiä esimerkiksi suurtarkkuuksisia nettivideoita katsottaessa, ja moni käyttäjä alkoi kaivata uudempaa laitetta. Mikäli purkutuki olisi säilytetty ajureissa, ei saman mittakaavan tarvetta uudempiin laitteisiin olisi.

## Fix-mod-hack = tee se itse!

Aloitin taipaleeni kulutuselektronikan parissa samoin kuin kaikki muutkin: käyttäjänä. Kun laite hajosi, se vietiin huoltoon. Kiinnostuin sähkölaitteiden toiminnasta jo nuorena, vaikka perheessäni ei niitä uskallettu itse huoltaa.



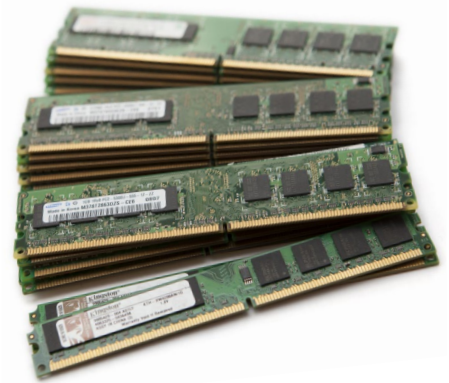
Dyykattuista koneista kasattu laskentafarmi. Eilispäivän tehokoneet tekevät hyvää tiimityötä. Työasemassa on esata-väylällä kiinni RAIDilla varustettu levypalvelin.

Ensimmäinen oma huolto oli VHS-videonauhuri, josta oli ajan saatossa kiuventunut kuminen kontaktipinta voimavälityksestä. Koska kyseistä osaa ei saanut käsikauppatavarana, piti sellainen tilata lähimmästä kodinkoneiden huoltoliikkeestä.

Seuraava merkittävä huolto oli television voittunut tasasuuntausmuuntaja. Ammattilaisilta kalasteltujen vinkkien avulla oma niksikokoelmani alkoi karttua. Kaverit ryhtyivät kantamaan ilmaiseen huoltooni polkupyöriään ja Mi-

nidisc-soittimiaan. Pikku hiljaa rupesin kasaillemaan tietokoneita, polkupyöriä ja kameroita. Dyykatut laitteet ovat hyviä harjoituskohteita, koska epäonnistuminen ei käy kukkarolle.

Yleisimpiä mekaanisesti kuluvia osia ovat remmit ja hihnat. Kierrätyskeskuksissa niitä löytää analogisista viihdelaitteista, kasetti- ja levysoittimista. Hihnoja käytetään myös pyykkikoneissa ja autoissa. Esimerkiksi auton nelitahtimoottorin venttiilien ajoitusta hoitaa nokka-akseli, jonka tulee olla synkronoitu kampiakse-



Käyttökelpoisia DDR2-muisteja löytää hylätyistä koneista kasapäin.



Valikoima laajennuskortteja. Edustalla kannettavan tietokoneen näyttöohjain (HD3650) sekä kannettavan PCI-E-väylän pöytäkonesovitin, taustalla WLAN-sovittimia.

liin ja mahdollisesti muihin apulaitteisiin kuten öljypumppuun. Ne löystyvät vääjäämättä, ja niiden vaihtaminen onkin huoltoliikkeiden säännöllistä leipätyötä.

### Suunniteltu vanheneminen

Monikansallisen Dupont-yrityksen hitti-tuote Nylon oli alkutaipaleellaan vähällä ajaa yrityksen vararikkoon. Nylon-sukkahousuja mainostettiin niiden kestävyydellä - ja kauppa kävi. Tuotemyynti kärsi kuitenkin jatkuvuuden puutteesta. Dupont ratkaisi ongelman tekemällä materiaalista heikompaa. Sen myötä teollisuuden piiriin syntyi uusi ala: suunniteltu vanheneminen ja esineen käyttöä ennakointi.

Käytäntöä perustellaan monesti sillä, että uusien tuotteiden myynnistä saatavat varat käytetään jatkokehitykseen. Tosiasia kuitenkin on, että suurin osa voitoista valuu kansainvälisten sijoituspankkiirien taskuihin. Muutaman vuosikymmenen kuluttua käsissämme on surullisen kokoinen vuori pari vuotta



Macbook Pro versio 1.1 vuodelta 2006 sekä versio 2.2. Jälkimmäisen muistimäärä rajoittuu kolmeen gigatavuun.

vanhoja tietokoneita, kameroita, ”liian pieniksi” käyneitä nestekidenäyttöjä, digibokseja ja reitittämiä.

### Elektroniikkajätteen ympäristövaikutukset

Teollistuminen ja massatuotannon synty toivat tavallisen kansan saataville kasapäin ennennäkemättömän hienoa tavaraa. Tavarataivaan taustalla pyörivät valmistusprosessit vaikuttavat ympäristöön kuitenkin paljon pidemmän aikaa kuin esineiden käyttöään verran. Uudet valmistusmenetelmät ovat vähentäneet raaka-aineiden tarvetta ja valmistuksesta koituvia ympäristöhaittoja, mutta samalla esimerkiksi kodinkoneiden käyttöikä on laskenut rajusti.

Uusien laitteiden tuottaminen vaatii paljon työtä ja työvaiheita. Siihen kuuluu kaikkea raaka-aineiden louhinnasta lopputuotteen valmistukseen ja markkinoinnin kautta myyntiin. Massatuotannon synnyttämät ongelmat ovat verrattain uusia. Vain muutama vuosikymmen sitten suurin osa kotitalousjätteestä oli myrkytöntä ja kompostoitavaa. Nyt uhka-

kuvana on se, että tuotantotaloutemme hedelmät putoavat tulevien sukupolvien syliin resurssipulana ja vakavina ympäristöongelmina.

Sähkö- ja elektroniikkaromu sisältää monia komponentteja, jotka ovat käyttöarvoltaan ja raaka-aineiltaan hyödyllisiä. Jopa yksittäisiä komponentteja voidaan käyttää uudelleen, ja niinpä elektroniikkajätteen hyödyntäminen raaka-aineena onkin kasvava teollisuudenala. Esimerkiksi arvometallien erotteluinen elektroniikkajätteestä edellyttää vahvojen kemikaalien käyttöä, joten kotikonstein se ei oikein onnistu. Sen sijaan teollisessa mittakaavassa se on ekonomisesti ja ekologisesti kannattavampaa kuin uuden arvometallin louhiminen.

Teknologiaa voidaan tehdä kestäväksi, mutta se on harvoin taloudellisesti kannattavaa. Tämä on valtava haaste varsinkin suurempien ja kalliimpien laitteiden kuten autojen kohdalla. Uuden auton valmistaminen saastuttaa enemmän kuin vanhan auton ajaminen loppuun asti, joten ympäristön kannalta on parempi, että autoja huolletaan ja ajetaan

mahdollisimman pitkään. Kulutusmielty-myksemme eivät kuitenkaan toimi ensisijaisesti järkiperusteilla, vaan markkina-koneistossa luodut kiillotetut mielikuvat ohjaavat käytöstämme.

### Tulevaisuus

Elektroniikkajätteen määrä on kasvussa. Uusien laitteiden hinnat ovat tippuneet ja huollot vähentyneet. Kaatopaikalla nököttää kasapäin resursseja. Vanhan skannerin tai tulostimen askelmoottori voisi löytää tiensä harrasteena rakennettavaan 3D-tulostimeen. 80-luvun järeä hifivahvistin todistaa, että komponentit voivat kestää aikaa. Vanhan vahvistimen äänenlaatu ja teho riittävät hyvin tänäkin päivänä. Uusi digitaalinen standardi edellyttäisi vain pienen mikropiirin ja liitännän vaihtamista, mutta laitevalmistaja markkinoi ja myy mieluummin kokonaan uuden laitteen. Tähän kuuluu paljon raaka-aineita ja energiaa.

Joudumme vääjäämättä turvautumaan vahvasti standardoituun modulaariseen teknologiaan, jossa apuvälineet suunnitellaan siten, että niiden päivittäminen ja huolto toteutuu mahdollisimman vähäisillä resursseilla. Valmistajan uusi vahvistinmalli saattaa tulevaisuudessa olla itse päivitettävä moduuli, joka sisältää uudet liitännät ja koodekit. Kehitteillä on materiaaleja, joista voidaan rakentaa vaikka syötäväksi kelpaavia akkuja ja näyttöjä.

Tulevaisuus saattaa olla valoisa, mutta se edellyttää asennemuutosta elektroniikkajätteen suhteen. Laitteita ei tule sysätä lyhyen käytön jälkeen kolmansien maiden ympäristöhaitaksi. Elektroniikkajäte on resurssivaranto, joka tulee hyödyntää. 3D-tulostamisen ja jyrkimisen mahdollisuudet varaosapuolella ovat huikeat. Suosittelemme perehtymistä laitteiden huoltoon, modaukseen ja virittelyyn!

## Dyykkauksen dinosaurusosasto

Teksti: Mikko Heinonen

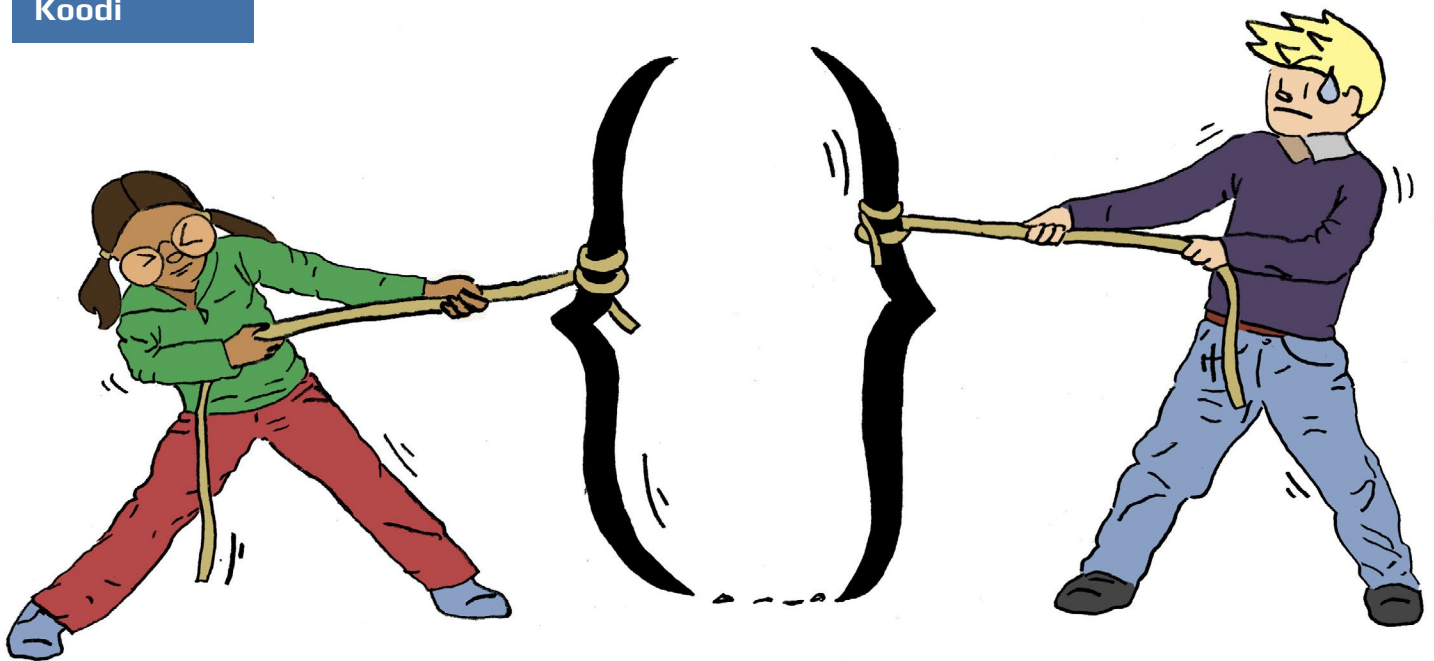
**K**uten kirjoittaja toteaa, on sähkö- ja elektroniikkaromun seasta mahdollista löytää käyttökelpoisia ja kohtuullisen modernia PC-rautaa. Mutta ei romulavojen anti tähän rajoitu. Myös kiinnostavaa vanhempaa tietokonetavaraa on helppo löytää: itsekin olen dyykkailut useita vanhoja pelikonsoleita,

suorittimia, muistipiirejä ja jopa klassikkokoneita, kuten Apple II:n.

Etenkin MS-DOS-aikakauden laitteissa on käynnissä varsinainen kato, sillä ne ovat jo poistuneet aktiivikäytöstä mutta eivät vielä nauti yleistä retroarvostusta. Silti niilläkin voi edelleen puuhastella kaikenlaista hauskaa. Esimerkiksi mukavan retroilutyöaseman saa korvaamalla kiintolevyn CF-kortilla, joka on jäänyt kamerakäytössä pieneksi. Koneella voi sitten katsella vanhoja DOS-demoja ja

pelata vaikka Doomia lähiverkossa kaverien kanssa. Lailisesti ilmaiseksi julkaistuja pelejä on netti pullollaan.

Kerrankin voi myös olla avoimesti nirso ja kelpuuttaa kokoonpanoonsa vain parhaat osat. Miten olisi huippunopea Pentium MMX L2-välimuistilla, emolevy täyteen RAMia, näyttökortiksi Tseng Labs ja äänille Gravis Ultrasound? Kaikki tämä löytyi ison firman romukontista ja kirpputorin ”miljoonalaatikosta” vain muutaman euron yhteishinnalla. 🐱



# Ikuisuuskyksymys: koodin muotoilu

*Kun ohjelmoidaan vähänkin isommalla porukalla, ennen pitkää joku nostaa aina kissan pöydälle koodin muotoilusta.*

Teksti: Jari Komppa Kuva: Annika Piironen

**M**onet koodin muotoiluun liittyvistä asioista ovat makuasioita, mutta pitkälti kyse on kuitenkin optimoinnista. Ei siis koodin nopeuttamisesta tai muusta tehostamisesta vaan ohjelmointiprosessin parantamisesta. Muotoilua optimoimalla voidaan saada aikaan parempaa ohjelmakoodia nopeammin.

Tietokoneiden alkupäivistä lähtien on ollut selvää, että suurin osa ohjelmoinnista on bugien metsästä eli koodin lukemista. Niinpä kannattaakin kirjoittaa koodia siten, että se on mahdollisimman helppolukuista. Kääntäjät kehittyvät jatkuvasti ja ovat pitkään olleet niin hyviä, että koodin kirjoittamisessa kannattaa mieluummin ajatella koodin luettavuutta kuin kääntäjää.

Käytössä on monenlaisia muotoilukäytäntöjä eli koodauskonventioita. Toisessa ääripäässä ovat käytännöt, joissa pyritään säästämään mahdollisimman paljon tilaa ja näppäinpainalluksia. Tällä tavoin pieneenkin tietokoneen ruudulla olevaan ikkunaan mahtuu näkyviin paljon koodia. Toisessa ääripäässä ovat käytännöt, jotka tuottavat ilmavaa, helppolukuista koodia.

## Neuvottelun tulos

Erään pelifirman alkuvaiheissa ohjelmajaporukka istui neuvottelutilassa muuttaman päivän riitelemässä kasaan muotoiluohjeen. Usein ihan pikkuseikoissakin piti valita useamman vaihtoehdon välillä, ja äänestykset voitti jokaisessa kohdassa

se, joka näytti vaativan enemmän työtä.

Muotoiluohjeet kannattaisi pitää parin kolmen sivun mittaisena, sillä muuten kasvaa riski, että niitä ei noudateta. Mainitun pelifirman ohje oli 22-sivuinen, mutta silti sitä seurattiin lähes pilkulleen. Lopputuloksena oli helposti ymmärrettävää lähdekoodia, ja työntekijät alkoivat käyttää samaa muotoilua myös joissakin harrastusprojekteissaan.

Tämä artikkeli ei käy koko muotoilukäytäntöä läpi mutta antaa esimerkkejä tärkeimmistä asioista. Mikäli tätä lukiesasi tunnet yhtäkkiä raivon nousevan, älä huoli, sillä nyt puhutaan asioista, joista joka tapauksessa kiistellään maailman tappiin asti.

Kielenä tässä on C tai C++, mutta samat asiat koskevat kaikkia C:n sukuisia kieliä.

## Sisennykset

Sisennykset ovat jostain syystä hyvin kiistelty seikka. Yleisimmin sisennyksissä käytetään kahta tai neljää välilyöntiä, joskin kahdeksaakin näkee välillä. Neljän välilyönnin puolesta puhuu lähinnä se, että se on Visual Studion oletusarvo.

```
kompleksi(parametri1, parametri2, parametri3,
           parametri4, parametri5, parametri6,
           parametri7, parametri8, parametri9);
```

```
kompleksi(parametri1, parametri2, parametri3,
           parametri4, parametri5, parametri6,
           parametri7, parametri8, parametri9);
```

Listaus 1. Sarkainasetusten muuttaminen muuttaa koodin muotoilua.

Eräessä virallisessa dokumentissa päädyttiin kolmen välilyönnin kompromissiin. Käytännössä sisennyksen suuruudella ei ole väliä niin kauan, kun siinä ollaan johdonmukaisia.

Sarkaimet toimivat eri tavoin eri tekstieditoreilla, joten sisennykset ja tasaukset kannattaa tallentaa tiedostoihin aina välilyönneillä. Listauksessa 1 on esimerkki, kuinka muotoilu voi räjähtää sarkainasetusten muuttuessa.

Ongelmat voi välttää käyttämällä sarkainmerkkejä pelkästään sisennykseen. Käytännössä tämäkin johtaa helposti siihen, että osa koodista seuraa oikeaa muotoilua ja osa ei. Helpoiten sotkusta pääsee määräyksellä, että kaikkialla käytetään välilyönnejä.

## Aaltosulkeiden paikka

Aaltosulkeilla määritellään koodilohko, ja siksi niiden kannattaa olla samalla tasolla sisennyksen kanssa. Aaltosulkeita kannattaa käyttää luettavuuden lisäämiseksi myös siellä, missä niiden käyttö on vapaaehtoista. Listauksessa 2 on esimerkki huonosta ja hyvästä käytännöstä.

## Etuliitteet

Muuttujien nimissä käytetään toisinaan etuliitteitä, jotka kertovat muuttujan ominaisuuksista. Kuuluisin näistä on niin sanottu unkarilainen notaatio, jossa muuttujien nimen alkuun merkitään muuttujatyyppi. Esimerkiksi muuttuja nimeltä pAddress on osoitin 32-bittiseen sanaan. Muun muassa Windowsin ohjelmointirajapinta käyttää unkarilaista notaatiota.

Etuliitteet ovat monella tavalla huono idea. Ne eivät helpota koodin ymmärtämistä juuri ollenkaan, ja muuttujien tyyppien vaihtuessa on vanha nimi korvattava kaikkialta koodista uudella. Vaihtoehtona on hyväksyä tilanne, jossa etuliite ei ole tae muuttujan tyyppistä. Surullisenkuuluisa esimerkki tästä on 32-bittisessä Windowsissa olevan WinMain-funktion parametri wParam, joka ei nimestään huolimatta ole 16-bittinen kokonaisluku. Microsoft ei vaihtanut muuttujan nimeä, koska se halusi säilyttää yhteensopivuuden 16-bittisen Windowsin kanssa.

Etuliitettä kannattaa mieluummin käyttää ilmaisemaan muuttujan näkyvyysaluetta, esimerkiksi seuraavan esimerkin tavoin:

```
aParametriMuuttuja
mJäsenMuuttuja
gGlobaaliMuuttuja
paikallinenMuuttuja
VAKIO_ARVO
LuokanTyyppi
```

Tämä käytäntö tekee koodista selvempää, koska koodi selittää itseään. Lyhyestäkin koodinpätkästä alkaa ymmärtää asioita, eikä termien merkityksiä tarvitse enää metsästellä sieltä täältä niin paljon. Vertaa listauksessa 3 olevia koodinpätkiä.

## Yksi rivi, yksi asia

Ilmavaa koodia on helpompaa lukea ja ymmärtää kuin tiivistä. Hyvä periaate on, että yhdellä rivillä ilmaistaan vain yksi asia. Valitettavasti varsinkin C-kielessä on hyvin helppoa kirjoittaa koodia, jossa yksi rivi tekee monia asioita. Niin on tehty esimerkiksi strcpy-funktiossa (listaus 4).

Kokenut ohjelmoija toki ymmärtää tiivistäkin koodia, mutta strcpy-funktion voi kirjoittaa huomattavasti helpommin ymmärrettäväksi. Jos muutaman sadan rivin mittainen funktio sisältää runsaasti vastaavanlaisia kikkoja, lopputulosta alkaa olla hyvin vaikeaa ymmärtää. Listauksessa 5 on strcpy:n koodi ilmavammin toteutettuna. Pätevä kääntäjä tuottaa kummastakin täysin samanlaisen lopputuloksen.

## Välilyönnit

Myös selvävä välylyöntien käyttö kuuluu koodin tyografiaan. Jotta koodi olisi mahdollisimman luettavaa, kannattaa kirjoittaa välylyönti operaattoreiden molemmin puolin, for- ja if-termien jälkeen sekä pilkkujen ja puolipisteiden jälkeen.

Listauksessa 6 näkyy välylyöntien vaikutus koodin luettavuuteen. Lausekkeet erottuvat hyvin toisistaan.

## Kommentit

Turhaa koodin kommentointia tulee välttää, mutta liian vähäinen kommentointi on yleisempi ongelma. Ikävimpiä esimerkkejä ovat lähdekoodin pätkät, joissa kirjoitetaan laitteiston rekistereihin mystisiä arvoja ilman yhtä ainuttakaan kommenttia, kun taas malloc-kutsulle on kirjoitettu kommentti "varataan muistia".

Kommentteja kannattaa käyttää aina, kun koodista on syntymässä jotakin, jota ei nopealla vilkaisulla ymmärrä. Tällainen tarve syntyy pakostakin silloin tällöin. Myös kiinteitä arvoja käytettäessä on hyvä selvittää, miksi arvo on valittu tai kuinka se on laskettu.

Pidemmissä kokonaisuuksissa kannattaa kertoa kommenttien avulla, mitä koodin on tarkoitus tehdä. Tällöin mahdollisia korjauksia tekevä henkilö voi tutkia, vastaavatko kommentit ja koodi toisi-

```
void bah() {
for(i=0;i<3;i++)
switch (x) {
case 3: if (foo) bar();
}
}
```

```
void bah()
{
for (i = 0; i < 3; i++)
{
switch (x)
{
case 3:
if (foo)
{
bar();
}
}
}
}
```

Listaus 2. Sisennyksen ja aaltosulkeiden yhteispeli huonosti ja hyvin toteutettuna.

```
if (foo)
{
bar = skaala * paino * painovoima;
}
```

```
if (aFoo)
{
bar = gSkaala * mPaino * PAINOVOIMA;
}
```

Listaus 3. Muuttujien näkyvyysaluetiedon voi sisällyttää muuttujan nimeen.

aan eli toimiiko ohjelma oikein.

Kommentteja kannattaa myös poistaa, jos ne eivät ole enää tarpeellisia. Joskus näkee hyvinkin epäolennaisia kommentteja, jotka ovat ehkä joskus olleet tärkeitä.

## Ja sitten käytännössä...

Mitä muotoilua sitten käytetäänkin, on tärkeää pysyä yhdenmukaisena. Laajemmissa projekteissa - varsinkin avoimen lähdekoodin maailmassa - törmää tilanteisiin, joissa saman projektin eri osat noudattavat eri käytäntöjä. Näissä tilanteissa on parasta pyrkiä samanlaiseen muotoiluun ja koodiin kuin aiemmin on samassa yhteydessä ollut, vaikka se ei noudattaisikaan omia mielilyöntejä. Yhtenäinen käytäntö voi nimittäin olla helpolukuisempi kuin tyylien sekamelska.

Mutta eipä sitten muuta kuin koodilemmisiin. Pidetään koodi kauniina! 🐱

```
char * strcpy(char *s1, char *s2)
{
char *o = s1;
while (*s1++ = *s2++);
return o;
}
```

Listaus 4. Tiiviiksi pakattua koodia strcpy-funktion tapaan.

```
char * strcpy(char *s1, char *s2)
{
char *o = s1;
char last = 1;
while (last)
{
*s1 = *s2;
last = *s1;
s1++;
s2++;
}
return o;
}
```

Listaus 5. Ilmavampi toteutus strcpy-funktiossa: yksi rivi, yksi asia.

```
if( x==(7 * *v-k) )
{
for( i=0;i<16;i++ )
{
boom(1.6,7,7.7,2,1.6);
}
}
```

```
if (x == (7 * *v - k))
{
for (i = 0; i < 16; i++)
{
boom(1.6, 7, 7.7, 2, 1.6);
}
}
```

Listaus 6. Sopivalla välylyöntien käytöllä lausekkeet erottuvat hyvin toisistaan.

## Koodaamisen kulttuureja

Teksti: Teemu Likonen

**C**-kääntäjä ei välitä, miten koodi on muotoiltu, joten ihmiset saavat tapella muotoiluohjeista keskenään. Kaikki tietenkin haluavat tehdä koodista helppolukuista ja helposti ylläpidettävää, mutta ongelmana on, että minun oikea tapani saattaa olla ihan eri kuin sinun. Otetaanpa esimerkki.

GNU-projektia varten on kirjoitettu dokumentti GNU Coding Standards, jonka tarkoituksena on yhtenäistää ohjelmointikäytäntöjä sekä ohjelmien toimintaa. Dokumentti ottaa kantaa myös koodin muotoiluun. Listauksessa 1 on dokumentista lainattu C-kielinen esimerkki. Sisennysporras on kaksi välilyöntiä, ja koodilohkoa ilmaisevat aaltosulkeet ovat omilla riveillään. Funktion nimi ja vasen sulje erotetaan toisistaan välilyönnillä.

Myös Linux-ytimen ohjelmoijilla on oma muotoiluohjeensa. Ohjeet perustuvat niin sanottuun K&R-tyyliin, joka tulee nimistä Brian Kernighan ja Dennis Ritchie. Heistä jälkimmäinen on C-kielen kehittäjä, ja yhdessä miehet kirjoittivat

kuuluisan oppaan The C Programming Language.

Linux-koodin muotoilu poikkeaa selvästi GNU:n suosituksista. Tyylikysymyksissä saattaa olla tunteitakin mukana, sillä Linuxin tyyliohjeen alussa kirjoitetaan ilmaiseksi oman näkemyksensä GNU:n ohjeista: "I'd suggest printing out a copy of the GNU coding standards, and NOT read it. Burn them, it's a great symbolic gesture."

Kun listauksen 1 koodi muutetaan Linux-ytimen tyyliohjeen mukaiseksi, saadaan listauksen 2 mukaista koodia. Sisennykset tehdään sarkainten avulla, ja sarkaimen leveys on kahdeksan merkkiä. Vasen aaltosulje kirjoitetaan usein samalle riville muun koodin kanssa. Jos if-rakenteen yhdessä haarassa tarvitaan aaltosulkeita, kirjoitetaan ne kaikkiin haarioihin.

Koodin vaihteleva muotoilu on vain yksi esimerkki siitä, että eri ohjelmoijayhteisöihin muodostuu erilaista kulttuuria. Siellä missä on joukko ihmisiä, on myös sosiaalisia normeja, ja niinpä ohjelmoijayhteisölläkin voi olla kaikenlaisia

sopimuksia hyvistä ohjelmointityyleistä, laadunvarmistuksesta ja ohjelman julkaisukäytännöistä. Yhteisön uusi jäsen tulee mukaan paitsi koodaamaan mutta myös uuteen koodauskulttuuriin.

```
if (x < foo (y, z))
  haha = bar[4] + 5;
else
{
  while (z)
  {
    haha += foo (z, z);
    z--;
  }
  return ++x + bar ();
}
```

Listaus 1. Esimerkki GNU-projektin suosituksista.

```
if (x < foo(y, z)) {
  haha = bar[4] + 5;
} else {
  while (z) {
    haha += foo(z, z);
    z--;
  }
  return ++x + bar();
}
```

Listaus 2. Esimerkki Linux-koodin muotoilusta.

Tekstissä mainitut ohjeet:

- GNU Coding Standards: <http://www.gnu.org/prep/standards/standards.html>
- Linux-ytimen tyyliohjeet: <http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/plain/Documentation/CodingStyle>

• KESTO  
48H

# somejam

**Missä: Nuorten toimintakeskus Happy**

Sörnäisten rantatie 31, 00500 Helsinki, 2. krs. Pelitalo

**Milloin: 14.-16.3.2014**

Tapahtuma alkaa 14.3. klo 18 ja päättyy 16.3. klo 18.

**Mitä:** Somejam on avoin tapahtuma, joka on suunnattu palvelumuotoilusta, palveluiden kehittämisestä ja innovatiivisesta ongelmanratkaisusta kiinnostuneille. 48 tunnin aikana luomme mahdollisimman konkreettisia keinoja parantaa teknologian avulla nuorille suunnattuja palveluita.

Lue lisää: [www.verke.org/somejam](http://www.verke.org/somejam) | Ilmoittautumiset: 9.3. mennessä

Tapahtuma on maksuton





## Kumpulassa kukkii

*Helsingin yliopiston Kumpulán kampuksen katolla, henkilökuntaa kummastuttaneessa majassa on viljelty chiliä jo parin vuoden ajan.*

Teksti: Kalle Viiri Kuvat: Jarmo Isotalo

Oppuvuodesta 2011 outo hätkyrä nousi Helsingin yliopiston Kumpulán kampuksen Exactum-rakennuksen katolle, henkilökunnan taukopaikkana käytetyn pienen kattoterassin viereen. Pääosin kierrätysvanerista ja pleksimuovista kasattu, vajaan kymmenen neliömetrin kokoinen mökki oli kasvihuone, jossa viljeltiin chiliä ympäri vuoden käyttämällä hyväksi palvelinten hukkalämpöä.

Projekti sai alkunsa vuonna 2010, kun kasvihuoneen perustajan, tutkijatohtori Mikko Pervilán työryhmä tutki palvelinten jäädyttämistä ulkoilmassa samaisella katolla. Yleensä merkittävä osa palvelinsalien käyttämästä sähköstä kuluu nimenomaan tietokoneiden tuottaman lämmön poistamiseen. Exactumin katolle rakennetun palvelinteltan avulla Pervilán työryhmä halusi tutkia, voisiko energiaa säästää jäädyttämällä palvelimia suoraan kylmässä ulkoilmassa.

Tulokset olivat lupaavia. Teltassa olevien palvelimien havaittiin toimivan luotettavasti jopa 20 asteen paukkupakkasilla.

### Hukkalämpö hyötykäyttöön

Vaikka palvelinten jäädytys ulkoilmassa on tehokasta, se ei ole ihanteellista. Lämpö menee nimittäin hukkaan. Kesällä 2011 projektia päätettiin laajentaa niin, että hukkalämpö käytettäisiin hyväksi. Aluksi lämmön käyttökohteeksi harkittiin saunaa. Tämä olisi kuitenkin vaatinut asiantuntemusta lämpöpumppujen ja eristysten osalta, ja kun sellaista ei löytynyt, ideasta jouduttiin luopumaan.

Sitten syntyi idea kasvihuoneesta. Talkoovoimin kasattu kasvihuone valmistui joulukuussa 2011 – mukavasti lumen ja pakkasen keskelle – ja palvelimet siirrettiin sen sisään. Huoneeseen asennettiin lisäksi useita antureita pitämään kirjaa kasvihuoneen lämpötilasta.

Valmiissa kasvihuoneessa alettiin kasvattaa chiliä. Kasvihuoneen rakentamisen yhteydessä Exactumin katolle asennettiin ”The Fifth Dimension – Green Roofs in Urban Areas” -projektin toimesta myös sammaleesta tehty viherkatto.

Palvelinten tuottama lämpö riitti pitämään chilit pääosin hengissä ensimmäisen talven yli, vaikka kasvihuoneen lämpötila putosi muutamaan otteeseen

lähelle nollaa. Kesällä kasvihuoneen ovet puolestaan avattiin ylikuumenemisen välttämiseksi. Chilisadosta tuli hyvä: yli 200 chilipaprikaa päätyi samassa rakennuksessa toimivaan Unicafe-ravintolaan opiskelijaruuan mausteeksi.

Kasvihuoneen palvelinprojekti päättyi syksyllä 2013, ja Pervilä sekä kasvihuonetta vapaaehtoisesti ylläpitäneet opiskelijat purkivat palvelimet ja anturit. Palvelinkoneet avattiin ja valokuvattiin tarkoituksena selvittää, kuinka paljon niihin kerääntyy likaa ulkoilmasta. Kasvihuone itsessään jätettiin paikoilleen ja lahjoitettiin tietojenkäsittelytieteen ainejärjestö TKO-äly ry:n käyttöön.

Ilman palvelinten tuottamaa lämpöä kasvihuoneessa voi viljellä vain kesäisin, mutta kasvukausi alkaa lämpimässä korpissa silti hieman aikaisemmin. Vielä ei ole tiedossa, mitä ainejärjestö aikoo viljellä korpissa seuraavaksi. 🌱

# Näkymättömänä netissä

*Netissä jokaisella on isovelji, joka tarkkailee, mutta joskus isoveljeäkin voi harhauttaa. Nyt otamme yksityisyyden perusasiat haltuun.*

Teksti: Santeri Tani Kuvat: Risto Mäki-Petäys, Santeri Tani

**T**ietämättömyys ajaa ihmisen toimimaan oman elämänsä tietovuotajana. Tiedon jakaminen internetissä on huomattavasti helpompaa ja nopeampaa kuin muualla, ja siksi myös yksityisasiat lipsahtavat helposti muiden luettavaksi. Tutulle henkilölle lähetetty yksityisviesti saattaa inhimillisen vahingon vuoksi päätyä sivullisille.

Suuryritykset ovat havainneet ihmisten tietämättömyyden ja suunnittelevat palvelunsa keräämään yksityistietojamme. Se on eräänlaista vakoilua, mutta muodollisesti se kuitenkin tapahtuu ihmisten omalla suostumuksella. Ihmiset suostuvat siihen, että esimerkiksi Facebook tallentaa kaiken ja oppii tuntemaan meidät. Tietojen avulla mainostajat saavat sopivia kohderyhmiä tuotteilleen, ja informaatiokauppa onkin valtava bisnes.

Suuryrityksillä on myös omat isoveljensä. Vuonna 2013 julkisuuteen vuotanut PRISM-ohjelma paljasti, että netin suurimmat yritykset ovat jakaneet käyttäjiensä tietoja Yhdysvaltojen hallituksen kanssa.

Yksityisyydellä ja sen suojelemisella on sekä sosiaalinen että tekninen ulottuvuutensa. Toisaalta on kyse siitä, mitä tietoja henkilö itse kertoo itsestään. Toisaalta laitteet ja ohjelmat saattavat levittää meistä tietoa. Järjestelmällisestä vakoilusta ja tarkkailusta on onneksi yhä mahdollista vapautua, tai ainakin siitä voi tehdä huomattavan vaikeaa. Tässä artikkelissa kerromme keinoja siihen.

## Ongelma penkin ja näppäimistön välissä

Tärkein työkalu yksityisyyden varjeluun ja tietoturvaan on terve maalaisjärki.

Aika pitkälle pääsee jo sillä, että toimii harkitusti.

Omaa nimeään, sähköpostiaan tai muita tietoja ei kannata kertoa joka paikassa. Nettipalvelujen käyttäjien kannattaa arvioida, mitä henkilötietoja palvelun toteuttamiseksi todella tarvitaan. Oikean nimen ja sähköpostin käyttö on monissa asioissa tietenkin suotavaa, mutta toisaalta ne tekevät käyttäjästä helposti jäljitettävän. Joissakin tilanteissa voi hyödyntää väliaikaisia sähköpostiosoitteita. Niitä tarjoavat esimerkiksi 10 Minute Mail ja Guerrilla Mail.

Tekstimuotoisen tiedon lisäksi myös ihan tavallinen valokuva voi sisältää tietoa, joka johtaa valokuvaajan jäljille. Tietenkin valokuvan kohteena oleva paikka voi olla tunnistettavissa, mutta kuvatiedosto voi sisältää muitakin tietoa. Kuvissa on niin sanottua EXIF-dattaa (Exchangeable image file format), johon usein lukeutuu kameran malli, kuvan ot-

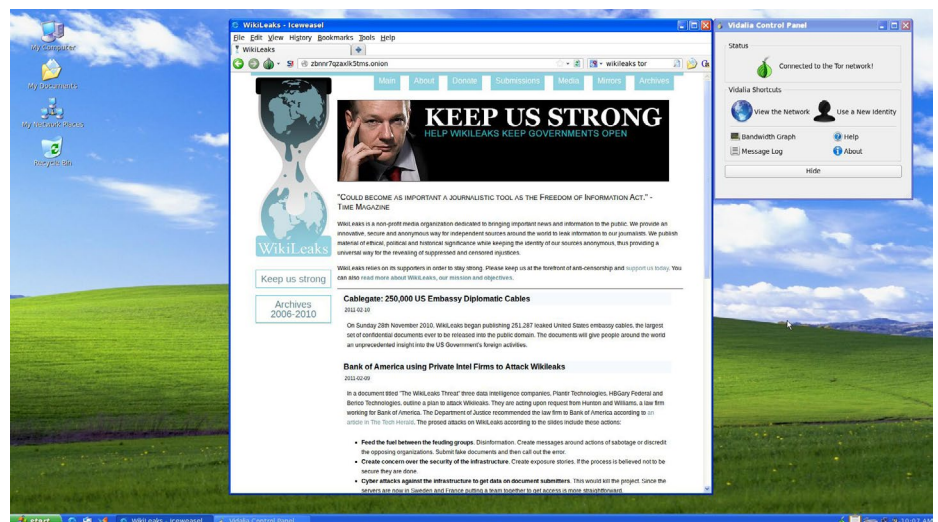
tohetki sekä GPS-koordinaatit. Niiden avulla kuvan ottajan sijainnin voi selvittää. Kuvien EXIF-dattaa voi poistaa tai muokata esimerkiksi Jhead-ohjelmalla.

## Vapaiden ohjelmistojen ilot

Tietokoneella tulisi käyttää vain tuttuja ja luotettavia ohjelmia. Ennen ohjelmien lataamista tulisi arvioida ohjelman tekijöiden sekä latauslähteen luotettavuus.

Käyttäjän vakoilemiseen tarkoitettuja käyttäjän tietoja keräävät ohjelmat ovat käytännössä aina suljettuja, eikä niiden sisäistä toimintaa voi kovin helposti tutkia. Monissa tapauksissa ohjelman käyttölisenssi myös kieltää ohjelman muokkaamisen, joten niihin ei ole edes luvallista vaikuttaa.

Yksityisyyden kannalta hyvä lähtökohta ovat käyttöjärjestelmät ja ohjelmat, joita kehitetään avoimen lähdekoodin periaatteella. Tällöin ohjelman lähdekoodi on vapaasti kenen hyvänsä tutkittavissa



Vidalia ja Tor-selain Windows XP:ksi naamioidussa Tails-käyttöjärjestelmässä.



ja ohjelmaan on vaikeampaa piilottaa koodia, joka ei kestä päivänvaloa.

Vapaiden ohjelmistojen käyttö voidaan ajatella eettisenä valintana. Järjestö nimeltä Free Software Foundation pyrkii edistämään vapaita ohjelmistoja ja tietoa niistä. Järjestön periaatteisiin kuuluu paitsi avoin lähdekoodi mutta erityisesti vapaus ohjelman käyttöön ja muokkaamiseen. Free Software Foundation on määritellyt neljä vapaan ohjelman periaatetta:

- Ohjelmaa saa käyttää mihin tahansa tarkoitukseen.
- Ohjelman toimintaa saa tutkia ja sitä voi mukauttaa omiin tarpeisiin. Tämän edellytyksenä on, että ohjelman lähdekoodiin on vapaa pääsy.
- Ohjelmaa voi levittää vapaasti, koska silloin siitä on hyötyä muillekin.
- Ohjelman parannettuja versioitakin saa vapaasti levittää. Näin koko yhteisö pääsee hyötymään parannuksista. Tämän edellytyksenä on, että lähdekoodiin on vapaa pääsy.

Free Software Foundation on julkaissut vapaiden ohjelmien levittämistä varten erilaisia ohjelmistolisenssejä. Tunnetuin niistä on GNU General Public License, josta onkin muodostunut hyvä vapaiden ohjelmistojen tunnusmerkki. Vapaita ohjelmistoja julkaistaan monilla muillakin lisensseillä, eivätkä kaikki edes määrittele vapautta samalla tavalla. Yksityisyyden ja luotettavuuden kannalta oleellisia ovat kuitenkin ohjelman kehitystyön sekä lähdekoodin avoimuus. Esimerkiksi GNU/Linux-nimellä tunnetut käyttöjärjestelmät ovat hyvin pitkälle vapaita, mutta niiden ohjelmavarastoista voi olla saatavilla myös suljettuja ohjelmia.

Tietoa Free Software Foundationin julkaisemista vapaiden ohjelmistojen lisensseistä: <http://www.gnu.org/licenses/>

On myös olemassa erityisesti tietoturvaan ja yksityisyyteen keskittyviä käyttöjärjestelmiä. Sellaisia ovat esimerkiksi Privatix, Tin Hat ja Pentoo, jotka ovat erittäin hyviä vaihtoehtoja yksityisyydestään huolestuneelle. Kehittyneemmille käyttäjille on virtualisointiin ja anonyymiverkkoihin perustuvia ratkaisuja kuten Tails, Janus VM ja Whonix, jotka ovat yksityisyydessä varsin tinkimättömiä.

## Tiedot suojaan

Käyttämällä vapaita ohjelmia, jotka on hankittu luotettavasta lähteestä, vältetään todennäköisesti vakoiluohjelmilta. Tietokoneen levyasemilla säilytetään kuitenkin paljon yksityisasioida, jotka voivat

## Foliohattuusio – kuinka saavutat yksityisyyden nirvanan

- Älä anna kenenkään muun käyttää tietokonettasi.
- Asenna ja käytä ainoastaan vapaita ohjelmistoja.
- Ennen käyttöjärjestelmän asennusta poista kaikki tiedot koneestasi esimerkiksi Darik's Boot and Nuke -levykkeellä.
- Suojaa kiintolevyysi ja käyttöjärjestelmäsi useilla salasanoilla. Muista, että BIOSin salasanan voi joissakin tapauksissa kiertää esimerkiksi poistamalla emolevyn pariston.
- Salaa kiintolevyysi, kotikansiosi ja kaikki tärkeät tiedostosi Truecryptillä, GNU Privacy Guardilla tai vastaavalla. Salaisimmat tiedot säilytät ainoastaan päässäsi.
- Vaikka koneellasi onkin GNU/Linux- taikka avoin BSD-käyttöjärjestelmä, käytä silti viruksetorjunta- ja palomuuriohjelmia kuten ClamAV:tä, Snortia ja Chkrootkitia.
- Poista tiedostot Bleach Bitillä tai Srm:llä, sillä normaali tiedostojen poistaminen merkitsee tiedot vain käyttämättömäksi. Itse tieto säilyy, kunnes sen päälle kirjoitetaan. Salatulla levyllä tämä ei ole tarpeellista.
- Tarkista ladattujen ohjelmistojen kryptografinen allekirjoitus tai muu varmenne käyttämällä GNU Privacy Guard-, md5sum-, sha1sum- tai vastaavaa ohjelmaa.
- Varmistu selaimesi jäljittämättömyydestä EFF:n user agent -vertailutyökalulla (<https://panopticklick.eff.org/>).
- Kaiken nettiliikenteen tulisi kulkea yksityisyyttä varjelevien verkkojen kuten Torin, i2p:n, Freenetin tai Hyperborian lävitse.
- VPN ja Tor: Yhdistä Toriin ainoastaan yksityisen erillisverkon (VPN) lävitse. Sellaisia tarjoavat esimerkiksi Ipredator ja Tor VPN, jotka myös hyväksyvät maksut nimettömästi Bitcoin-rahalla. Sinun tulee luottaa yksityisen erillisverkkosi tarjoajaan vähintään yhtä paljon kuin verkkosi palveluntarjoajaan. Välityspalvelimet vuotavat usein IP-osoitteesi eivätkä ole salattuja.
- Whonix: Peitä IP-osoitteesi käyttöjärjestelmän sisällä suoritettavilla virtuaalikäyttöjärjestelmillä, joista toinen toimii nettiliikenteen mahdollistavana reitittimenä. Tällöin edes root-käyttäjän oikeuksilla varustettu haittaohjelma ei voi selvittää käyttäjän oikeaa IP-osoitetta.
- Tails: Käytä dvd-levyltä tai usb-muistilta käynnistettävää Tails-käyttöjärjestelmää, kun haluat käyttää internetiä muiden tietokoneella.
- Älä ikinä käytä langatonta verkkoa.
- Älä käytä selaimen Javascriptia, älä pidä päällä selaimen "do not track" -asetusta äläkä hyväksy evästeitä. Käytä Jon Do Foxin yksityisyysprofiilia Iceweaselissasi.
- Jos haluat välttämättä käyttää yhteisöpalveluja, käytä Diasporaa tai Friendicaa.
- Suorita internet-haut käyttäen Ixquickia, Startpagea tai hajautettua Hyperboriaa.
- Sulje internet-yhteys, kun et tarvitse sitä.

muulla tavalla päätyä väärin käsiin. Entä jos tietokone varastetaan?

Omien tietojen suojaamiseksi on olemassa erilaisia salausohjelmia. Salaus voidaan toteuttaa kiintolevyn tiedostojärjestelmän tasolla, jolloin levyn kaikki sisältö tallennetaan automaattisesti salatussa muodossa. Toisaalta tiedostojärjestelmän sisällä yksittäiset tiedostot voidaan salata. Tällöin tiedoston salaamisen saattaa joutua hoitamaan käsityönä.

Hyviä salaustyökaluja ovat GNU Privacy Guard (lyh. GPG) ja Truecrypt. Linux-ytimessä itsessään on salaustoimintoja, ja kiintolevyn tiedostojärjestelmä voidaan salata sen avulla. Joidenkin GNU/Linux-käyttöjärjestelmien asentamisen yhteydessä voi valita, että kiintolevylle luodaan salattu osio, jonka sisään kaikki asennetaan.

## Internet

Internetissä liikkuva käyttäjä jättää suuren määrän jälkiä itsestään. Nettiin

kytketyillä laitteilla on IP-osoite, jota käytetään laitteiden yksilöimiseen ja datan reitittämiseen. Internet-yhteyden ja -käytön aikana IP-osoite tallentuu moniin paikkoihin, esimerkiksi verkkokauppojen ja keskustelufoorumeiden tietokantoihin. Osoitteen perusteella voidaan helposti selvittää käyttäjän internet-yhteyden tarjonnut yritys sekä sijainti ainakin suurin piirtein. Poliisi ja valtiolliset tiedustelupalvelut voivat saada palveluntarjoajien kirjanpidosta (tai vakoilemalla) tarkemat tiedot henkilöstä, joka on ollut IP-osoitteen takana.

PRISM-ohjelma paljasti lähes kaikkien suurien nettipalvelujen tietoliikenteen kulkevan Yhdysvaltojen kansallinen turvallisuusviraston (National Security Agency, NSA) kourien lävitse. Suositut palvelut kuten Facebook, Youtube ja Google saattavat käsittää valtaosan peruskäyttäjän verkkoliikenteestä, joten niiden käyttäjä on jatkuvasti NSA:n näkökentässä.

<b>general.useragent.override</b>	Mozilla/5.0 (Windows NT 6.1; rv:10.0) Gecko/20100101 Firefox/10.0
<b>general.appname.override</b>	Netscape
<b>general.appversion.override</b>	5.0 (Windows)
<b>general.oscpu.override</b>	Windows NT 6.1
<b>general.platform.override</b>	Win32
<b>general.productSub.override</b>	20100101
<b>general.buildID.override</b>	0
<b>intl.accept_languages</b>	en-us,en;q=0.5
<b>network.http.accept.default</b>	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
<b>network.http.accept-encoding</b>	gzip, deflate

Taulukko 1. Näin Firefox saadaan näyttätymään suosittuna perusselaimena.

Valtaosa verkkoliikenteestä on salaamatonta. Internet-yhteyttä ylläpitävä yritys ja mahdollisesti jokin muukin osapuoli pystyy valvomaan verkkoliikennettä, mikäli tiedonsiirtoa nettipalvelun ja käyttäjän välillä ei ole salattu tai internet-yhteyttä ei ole muodostettu virtuaalisen erillisverkon (VPN) lävitse.

Netin selailuun kannattaa käyttää avoimen lähdekoodin ohjelmaa kuten Firefoxia tai sen sisarversioita, Iceweaselia tai Icecatia. Jos haluaa Chrome-tyyppisen selaimen, voi käyttää avointa Chromium-selainta.

Selaimen asetuksista kannattaa poistaa käytöstä sivuhistoria sekä evästeet (cookies). Mikäli tarvitsee ja haluaa säilyttää selaimessaan evästeitä, Firefoxin lisäosa Better Privacy on sopiva työkalu niiden hallintaan. Joissakin selaimissa on "do not track" -asetus, jolla voi pyytää nettisivustoja olemaan seuraamatta käyttäjän tekemisiä. Ehkä se joskus toimii, mutta toisaalta sen avulla sivusto voi helpommin tunnistaa käyttäjän netti-

selaimen.

Firefoxissa selaimen ja käyttöjärjestelmän tunnistettavuutta voi vähentää asettamalla selaimen user agentin mahdollisimman yleismaailmalliseksi. Tällä tavoin nettiselain tavallaan piiloutuu suureen käyttäjämäärään eikä jätä itsensä yksilöllisiä vihjeitä.

1. Kirjoita osoiteriville "about:config", niin pääset muokkaamaan asetuksia.
2. Lisää uusi asetus "general.useragent.override".
3. Anna sen arvoksi mahdollisimman yleinen user agent, esimerkiksi "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-GB; rv:1.8.1.6) Gecko/20070725 Firefox/2.0.0.6".

Näillä asetuksilla selain kertoo palvelimelle, että käytössä on Windows XP ja selaimena Firefox 2.0.0.6. Mikäli haluat olla erityisen varovainen, voit myös lisätä taulukon 1 arvot selaintunnistuksen estämiseksi.

Nettisivut voivat ajaa selaimessa eri-

laisia skriptejä, joista jotkut tarkkailevat käyttäjän toimintaa tai ovat jopa puhtaasti haittaohjelmia. Firefoxin Noscript tai Chromiumin Scriptsafe ovat lisäosia, joilla käyttäjä pääsee itse hallitsemaan skriptien suorittamista.

Yhteys nettisivulle kannattaa tietenkin muodostaa salattuna, aina kun se vain on mahdollista. HTTPS Everywhere on Electronic Frontier Foundationin kehittämä lisäosa, joka tarkistaa, onko sivustolle mahdollista ottaa yhteys salatun https-protokollan avulla.

Virallisen Javan sijasta kannattaa käyttää vapaata Icedtea-toteutusta. Flashin sijasta kannattaa käyttää vapaata Gnashia. Hakukoneena kannattaa käyttää Googlen sijasta Ixquickia, Startpagea tai Duck Duck Gota.

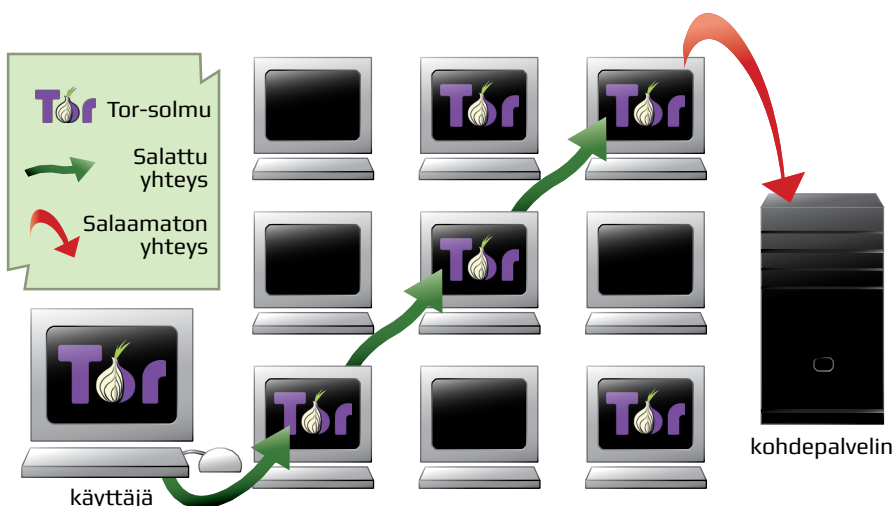
Kaiken tämän jälkeenkin nettiselaimen käyttö tekee käyttäjästä melko haavoittuvaisen. Yksityisyydestään huolestunut henkilö käyttää selainta ainoastaan sellaisiin asioihin, joissa on muutenkin pakko esiintyä oikealla nimellä.

## Tor

Anonyymiverkko Tor on saanut viime aikoina valtavasti suosiota. Oikein käytettynä se on varsin turvallinen. Tor syntyi Yhdysvaltojen armeijan tutkimuslaitoksen (United States Naval Research Laboratory, NRL) rahoittamana valtiotason kommunikaatiotyökaluna. Vuonna 2004 Tor alkoi saada rahoitusta elektronisia oikeuksia puolustavalta Electronic Frontier Foundationilta ja muuttui vapaaksi ohjelmaksi. Nykyään Tor on itsenäinen projekti, jota rahoitetaan vapaaehtoisilla lahjoituksilla. Suurimmat rahamäärät tulevat Yhdysvaltojen ja Ruotsin hallitukselta.

Tor-reitittimet ovat suurimmaksi osaksi yksityisten ihmisten ylläpitämiä. Aktiivisia Tor-reitittämiä on noin viisi tuhatta, mikä on aivan liian vähän neljälle miljoonalle päivittäiselle käyttäjälle. Reitittimien pieni määrä suhteessa käyttäjämäärään johtaa siihen, että tiedonsiirto Tor-verkon kautta on melko hidasta.

Tor-ohjelma käynnistää tietokoneelle socks-välityspalvelimen, jonka kautta tieto siirretään. Verkossa Tor käyttää onion routing -salaustekniikkaa, jossa tieto siirretään kolmen Tor-reitittimen eli solmun kautta. Käyttäjän tietokone ottaa yhteyden ensimmäiseen solmuun, eikä se näe muuta kuin ensimmäisen solmun IP-osoitteen. Ensimmäinen solmu näkee ainoastaan käyttäjän sekä toisen solmun osoitteen, ja toinen solmu tietää ainoastaan ensimmäisen solmun sekä ketjun viimeisen solmun osoitteen.



Tor-verkon rakenne.



## Congratulations. Your browser is configured to use Tor.

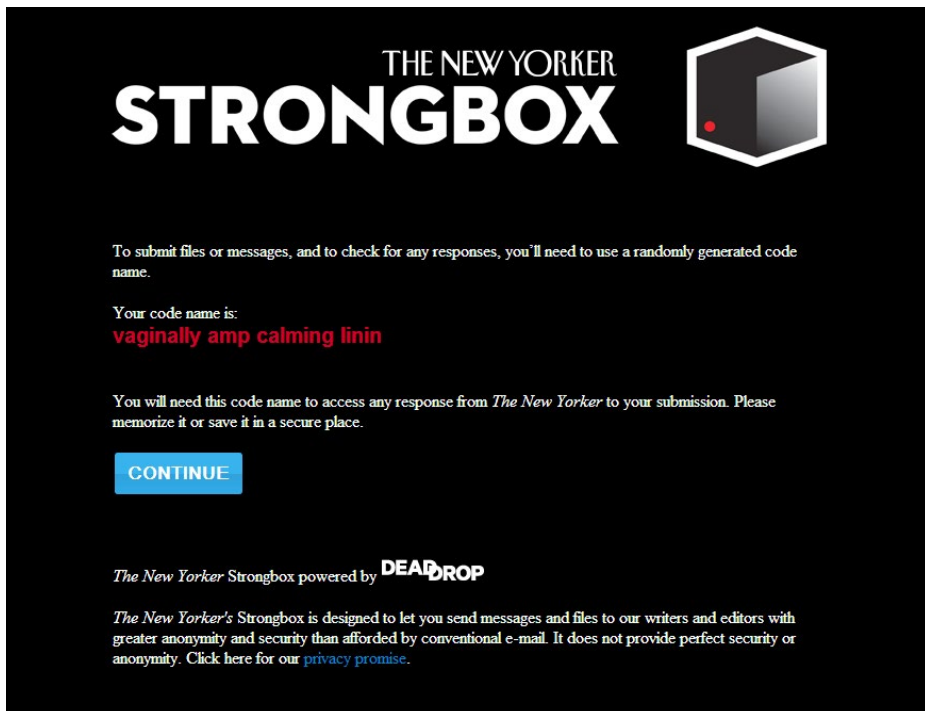
Please refer to the [Tor website](#) for further information about using Tor safely. You are now free to browse the Internet anonymously.

Your IP address appears to be:

This page is also available in the following languages:

[العربية](#) (Arabi) [Burmese](#) [česky](#) [dansk](#) [Deutsch](#) [Ελληνικά](#) (Ellinika) [English](#) [español](#) [Estonian](#) [فارسی](#) (Färsi) [suomi](#) [français](#) [Italiano](#) [日本語](#) (Nihongo) [norsk](#) (bokmål) [Nederlands](#) [polski](#) [Português](#) [Português do Brasil](#) [română](#) [Русский](#) (Russkii) [ไทย](#) [Türkçe](#) [українська](#) (Ukrainska) [Tiếng Việt](#) [中文](#) (Zhongwen)

Torin toiminnan virallinen tarkistussivusto <https://check.torproject.org/>.



**THE NEW YORKER STRONGBOX**

To submit files or messages, and to check for any responses, you'll need to use a randomly generated code name.

Your code name is:  
**vaginally amp calming linin**

You will need this code name to access any response from *The New Yorker* to your submission. Please memorize it or save it in a secure place.

**CONTINUE**

The New Yorker Strongbox powered by **DEADROP**

The New Yorker's Strongbox is designed to let you send messages and files to our writers and editors with greater anonymity and security than afforded by conventional e-mail. It does not provide perfect security or anonymity. [Click here for our privacy promise.](#)

The New Yorkerin palvelu nimettömille uutisvihjeille ja tiedoille.

Tor-verkossa solmujen väliset yhteydet ovat salattuja, lukuun ottamatta viimeisen solmun ja kohdepalvelimen välistä yhteyttä. Kukaan osapuolista ei näe koko ketjua. Esimerkiksi kohdepalvelin ei tiedä, kuka on yhteyden alun perin muodostanut, koska yhteys näkyy tulevan vain yhdestä Tor-verkon solmusta.

Ehkä helpoimmin Tor-verkkoa pääsee käyttämään, kun lataa Tor-projektin nettisivuilta Tor Browser Bundle -selainpaketin. Paketissa on mukana Firefox-selain, joka on valmiiksi asetettu toimimaan Tor-verkossa. Mukana on myös Vidalia-ohjauspaneeli, jolla voi säätää verkon asetuksia.

Anonyymien netinkäytön lisäksi Tor-verkkoon voi myös perustaa piilopalveluja, joihin saa yhteyden ainoastaan Tor-verkon kautta. Tor ja sen piilopalvelut ovat siis eräänlainen internetin sisällä

oleva piiloverkko. Piilopalvelujen osoitteet ovat .onion-päätteisiä.

Piratebay on julkaissut sensuurin kiertämiseksi oman Piratebrowserinsa, joka on lähes identtinen Tor Browser Bundlen kanssa mutta tekeekin kolmen reititinloikan sijasta vain yhden. Piratebrowser ei siis suojele käyttäjän yksityisyyttä kovin hyvin, koska yhteyden välissä oleva solmu näkee sekä käyttäjän tietokoneen että kohdepalvelimen.

### Hyödyllisiä Tor-osoitteita

- Yksi vanhimmista piilopalveluista listaavista sivustoista, Tor-projektin ylläpitämä: <http://eqt5g4fuenphqinx.onion/>
- Kotisuomalainen laillisia .onion-sivustoja indeksoiva hakukone: <https://ahmia.fi/>
- Duck Duck Go -hakukoneen Tor-versio: <https://3g2upl4pq6kufc4m.onion/>
- Yksityisyyteen keskittyvä sähköpostipalvelu UR55 Mail: <http://f3ljvgyyujmnnfvi.onion/>

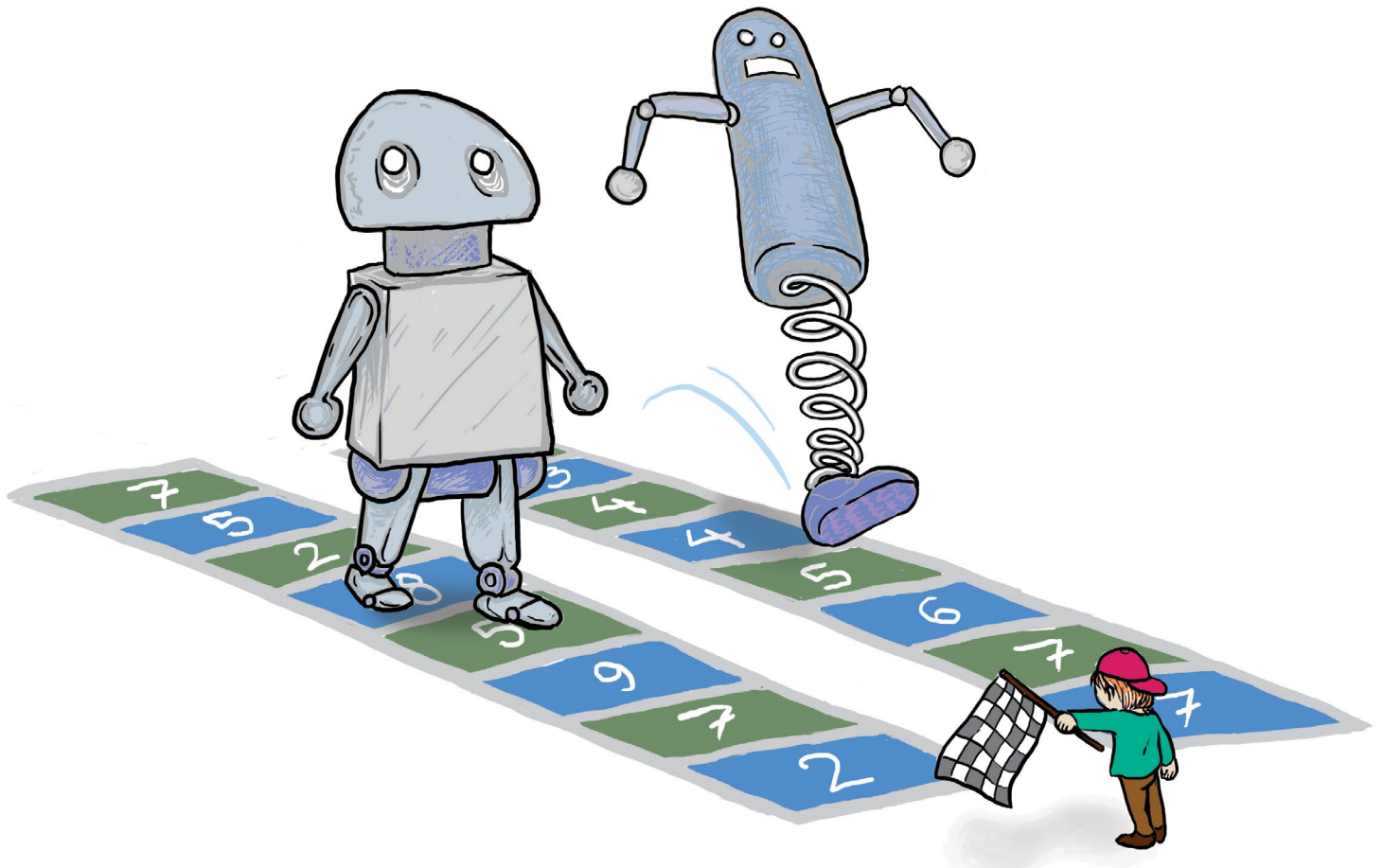
## Miksi Tor – oletko sananvapaustaistelija vai terroristi?

Tor on saanut sekä hyvää että huonoa julkisuutta. NSA-tietovuotaja Edward Snowden käytti Toria ja sähköpostipalvelu Lavabitia, kun lähetti lehdistölle tietoja valtion PRISM-vakoiluohjelmasta. Palvelut kuten Global Leaks ja New Yorker Strongbox vahvistavat Torin asemaa sananvapaustyökäluna, koska niiden kautta voi turvallisesti lähettää uutistietoa maailmalle.

Joidenkin valtioiden hallitukset säätelevät ulkomaille meneviä internet-yhteyksiä hyvin tarkkaan. Silloin Tor saattaa muodostua tärkeäksi kanavaksi arkaluontoisen tiedon välittämiseen. Tosin Tor-yhteydetkin on mahdollista tunnistaa ja estää.

Anonymiteetille ja sananvapaudelle on aina väärinkäyttäjiä: piilosivustoilla on jaettu teratavukaupalla lapsipornoa sekä myyty huumeita. Salaisten palvelujen ja niiden käyttäjien jäljittäminen on vaikeaa, mutta kyberturvallisuuteen panostavan valtion tiedustelupalvelu saattaa siinäkin onnistua. Esimerkiksi Yhdysvaltojen keskusrikospoliisi (Federal Bureau of Investigation, FBI) on asentanut laittontaa materiaalia jakavalle palvelimelle Javascriptia hyödyntävän ohjelman, joka vakoili palvelimen käyttäjiä.

Tor ja muut salaustekniikat voivat olla yhtä lailla myönteisen sananvapauden kuin rikollistenkin työkaluja. Ne ovat vain tekniikkaa, joka ei itsessään sisällä moraalista kannanottoa. Aina ei edes ole selvää, kummasta on kyse. Sama henkilö voi olla toisille sananvapauden puolustaja ja toisille rikollinen. 🐼



# Maailman vaikein algoritmi

*Moni koodari on yrittänyt toteuttaa binäärihaun, mutta tuskin kukaan on onnistunut siinä. Tämän jutun luettuasi voit liittyä harvojen ja valittujen joukkoon.*

Teksti: Antti Laaksonen Kuva: Mitol Berschewsky

**B**inäarihaku on ohjelmoinnin teorian tunnetuimpia algoritmeja. Sen avulla voi esimerkiksi tarkistaa tehokkaasti, esiintyykö annettu luku järjestetyssä taulukossa. Toisaalta binäärihaku on tullut tunnetuksi myös siitä, että algoritmin toteuttaminen virheettömästi on lähestulkoon mahdotonta. Näin asia onkin, jos algoritmin toteuttaa perinteisellä menetelmällä. Toteuttaminen muuttuu kuitenkin lastenleikiksi, kun valitaan toisenlainen lähestymistapa.

## Binäärihaulla olisit jo perillä

Oletetaan, että taulukko  $T$  sisältää  $n$  lukua ja että tehtävänä on tarkistaa, esiintyykö luku  $k$  taulukossa. Yksinkertaisin algoritmi tähän tehtävään on *lineaarihaku*, joka käy läpi taulukon luvut yksi kerrallaan. C-sukuisella kielellä lineaarihaun voisi toteuttaa seuraavasti:

```
for (int i = 0; i < n; i++)
    if (T[i] == k)
        return true;
return false;
```

Tarkistusta ei voi tehdä lineaarihakua tehokkaammin, jos taulukon alkioit voivat olla sekaisin. Tällöin etsittävä alkio voi olla missä tahansa taulukon kohdassa, ja algoritmin täytyy pahimmassa tapauksessa käydä läpi kaikki alkioit. Tilanne kuitenkin muuttuu, jos taulukon sisältö on järjestetty. Tällöin *binäärihaku* on ylivoimaisen tehokas algoritmi etsimiseen.

Perinteinen binäärihakuun liittyvä vertauskuva on nimen etsiminen puhelinluettelosta. Aluksi puhelinluettelo avataan keskeltä ja katsotaan, mikä nimi siinä kohdassa on. Nyt haun voi rajoittaa luettelon alkuosaan tai loppuosaan riippuen siitä, mikä nimi on luettelon keskellä. Hakualetta puolitetaan tällä tavalla yhä uudestaan, kunnes alue on niin pieni, että nimi löytyy helposti. Idea on hauska, mutta ikävä kyllä binäärihaun koodaaminen bugittomasti on tämän pohjalta vaikeaa.

Helpompi lähestymistapa on ajatella binäärihakua lineaarihaun tehostuksena. Ideana on siirtyä taulukossa eteenpäin hyppäyksin. Ensin hypyn pituus on  $n/2$

indeksiä, sitten  $n/4$  indeksiä, sitten  $n/8$  indeksiä jne. Haun alussa voidaan tehdä pitkiä loikkia, mutta vauhti hidastuu, kun tullaan lähemmäksi etsittävää lukua. Tuloksena on seuraava koodi:

```
int i = 0;
for (int b = n / 2; b >= 1; b /= 2)
    while (i + b < n && T[i + b] <= k)
        i += b;
return T[i] == k;
```

## Vaikeampaa kuin arvaisi

Binäärihaun toteuttaminen on yllättävän vaikeaa. Jon Bentley antaa tästä esimerkin kirjassaan *Programming Pearls* (1986). Bentley suoritti useita kertoja kokeen, jossa ammattiohjelmoijat saivat tehtäväkseen toteuttaa binäärihaun puhelinluetteloesimerkin perusteella. Muutaman tunnin kuluttua lähes kaikki ilmoittivat saaneensa algoritmin valmiiksi, mutta tarkemman tutkimuksen jälkeen osoitettiin, että vain noin 10 % toteutuksista oli toimivia. 20 vuotta Bentley'n kirjan julkaisun jälkeen paljastui, että myös kirjassa esitetyssä binäärihaun toteutuksessa on bugi!

Muuttuja  $b$  määrittää, kuinka suuria hyppyjä haussa tehdään. Jokaisen hypyn pituuden kohdalla liikutaan niin monta hyppyä eteenpäin kuin mahdollista menemättä taulukon oikean reunan yli tai sellaiseen lukuun, joka on etsittävää lukua suurempi. Kuten aiemminkin, muuttuja  $i$  on taulukkoon osoittava indeksi. Haun päätteeksi  $i$  osoittaa kohtaan, jossa haettava luku esiintyy taulukossa, mikäli sellainen kohta on olemassa.

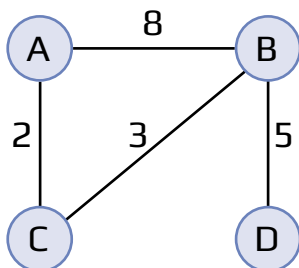
Matemaattisesti ilmaistuna lineaarihaku tarkistaa pahimmillaan  $n$  taulukon kohtaa, kun taas binäärihaku tarkistaa vain noin  $\log_2 n$  kohtaa. Binäärihaun askelten logaritminen määrä johtuu siitä, että hypyn pituus puolittuu jokaisessa vaiheessa. Lisäksi tiettyä hypyn pituutta vastaavia askelia tehdään käytännössä korkeintaan kaksi. Ero  $n:n$  ja  $\log_2 n:n$  tarkistuksen välillä on todella merkittävä, kun  $n$  on suuri luku. Esimerkiksi jos  $n$  on miljoona,  $\log_2 n$  on vain noin 20.

### Mitä sillä tekee?

Osaamme nyt koodata toimivan binäärihaun, mutta mihin sitä voisi käyttää? Yllättävää kyllä, binäärihakua käytetään nykyään harvoin järjestetystä taulukosta hakemiseen, sillä useimmissa ohjelmointikielissä on valmiina paljon monipuolisempia tietorakenteita. Esimerkiksi C++:ssa voi käyttää `std::set`-luokkaa, joka tukee tehokkaan haun lisäksi myös tehokasta alkion lisäämistä ja poistamista.

Binäärihaulla on kuitenkin muitakin sovelluksia. Tutkitaan vaikkapa tieverkkoa, jossa on kaupunkeja sekä kahden kaupungin välisiä teitä. Jokaisesta tienpätkästä tiedetään, paljonko polttoainetta sillä ajaminen kuluttaa. Autot voivat tankata kaupungeissa mutta eivät tieosuuksilla.

Minkä kokoinen polttoainesäiliö autossa täytyy vähintään olla, jotta sillä voi ajaa mistä vain kaupungista mihin tahansa toiseen kaupunkiin?

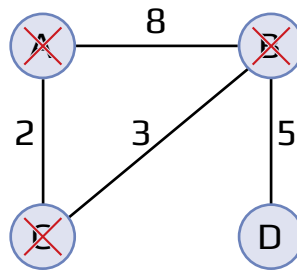


Kuva 1. Polttoaineongelman lähtöasetelma.

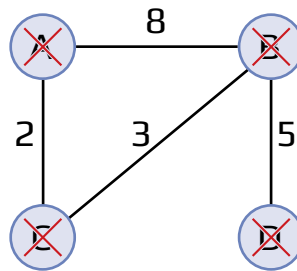
Kuvassa 1 on esimerkki tieverkosta. Jokainen kuvan ympyrä vastaa kaupunkia, jotka on merkitty tunnuksin A, B, C ja D. Kaupunkien väliset tiet on merkitty viivoihin. Esimerkiksi kaupungista B pääsee kaupunkiin C kuluttamalla kolme litraa polttoainetta.

Tehtävän ratkaisussa kannattaa ensin miettiä yksinkertaisempaa ongelmaa: jos polttoainesäiliön koko on  $x$  litraa, onko mahdollista ajaa minkä tahansa kahden kaupungin välillä?

Ongelma ratkeaa näin: Valitaan mikä tahansa kaupunki lähtökaupungiksi ja merkitään se ruksilla. Sitten merkitään kaikki kaupungit, joihin lähtökaupungista pääsee  $x$  litralla polttoainetta. Tämän jälkeen merkitään vastaavasti kaikki kaupungit, joihin pääsee viimeksi merkityistä kaupungeista. Samaa toistetaan niin kauan kuin uusia kaupunkeja tulee mukaan. Jos lopussa kaikki kaupungit on merkitty,  $x$  litraa riittää minkä tahansa kahden kaupungin välillä ajamiseen.



Kuva 2. Mihin pääsee C:stä kolmen litran säiliöllä?



Kuva 3. Viiden litran säiliö riittää kaikkialle.

Luodaan `riittaa(x)`-funktio, joka palauttaa `true`, jos  $x$  litran säiliö riittää kaikkien kaupunkien välillä, ja muuten `false`. Jos tehtävän ratkaisu on  $r$  litraa, `riittaa(x)` palauttaa `false` aina, kun  $x$  on  $r-1$  tai vähemmän, ja `true` aina, kun  $x$  on  $r$  tai enemmän.

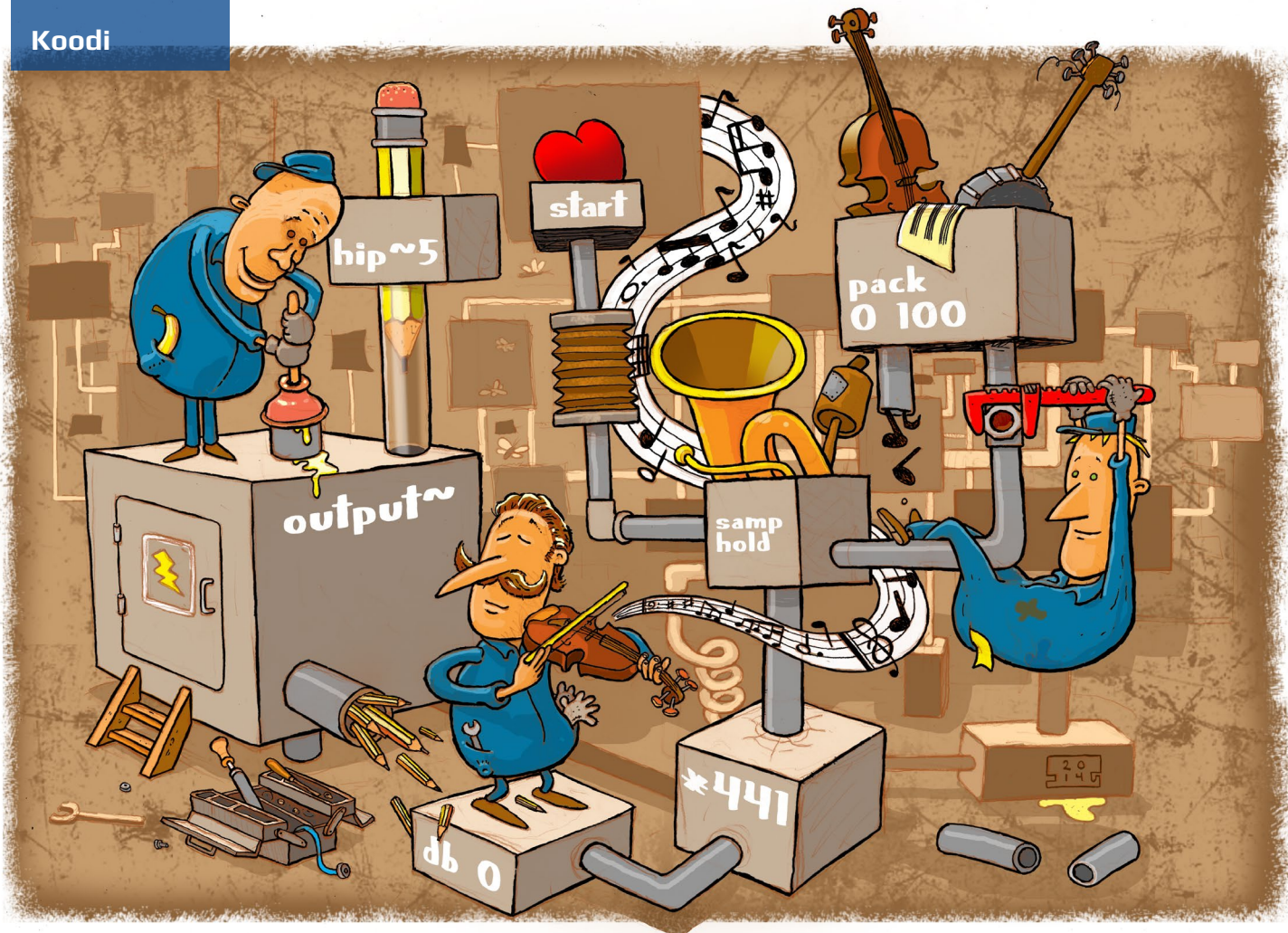
Funktion arvoissa on siis yksi muutospaikka, jonka vasemmalla puolella jokainen arvo on `false` ja oikealla puolella jokainen arvo on `true`. Lisäksi tiedämme,

että kaupungista toiseen pääsee ainakin silloin, kun polttoainesäiliö riittää kaikille tieosuuksille. Näiden ominaisuuksien ansiosta tehtävän voi ratkaista tehokkaasti binäärihaulla funktiota `riittaa(x)` käyttäen:

```
int v = 0;
for (int b = s / 2; b >= 1; b /= 2)
    while (!riittaa(v + b))
        v += b;
return v + 1;
```

Muuttujassa  $s$  on tieverkon vaativimman tieosuuden polttoainekulutus. Tämä arvo on luonteva valinta binäärihaun ylärajaksi. Koodi laskee muuttujaan  $v$  suurimman polttoainemäärän, joka ei riitä minkä tahansa kahden kaupungin välillä ajamiseen. Tehtävän vastaus on siten yhden suurempi kuin tämä arvo. Binäärihaun ansiosta funktiota `riittaa(x)` täytyy kutsua vain noin  $\log_2 s$  kertaa.

Vastaavaa menetelmää voi soveltaa aina, kun etsitään lukua ja osataan tunnistaa, onko jokin annettu luku pienempi vai suurempi kuin etsitty luku. Binäärihaun ansiosta ei tarvitse käydä läpi kaikkia mahdollisia arvoja vaan logaritminen määrä riittää. Algoritmista tulee siis tehokas. 🐢



## Pure Data – Äänen ja kuvan kieli

*Multimediaohjelmointi on helppoa ja hauskaa. Pure Data -kielen avulla käsket tietokoneen soittamaan oman pillisi mukaan.*

Teksti: Heikki Ketoharju Kuvat: Tapio Lehtimäki, Heikki Ketoharju

**M**ultimedian tuottamiseen on olemassa paljon erilaisia ohjelmia, ja media-alalla lähes kaiken työn voi nykyään tehdä tietokoneella. Luova ihminen saattaa kuitenkin jossain vaiheessa törmätä valmiiden ohjelmien rajoituksiin ja alkaa toivoa joustavampia työkaluja. Olisi hauskaa rakentaa omia tietokoneohjelmia, jos ohjelmointi vain ei olisi niin hankalaa!

Pure Data on helppo ohjelmointikieli, joka on suunniteltu multimediakäyttöä varten. Sillä saa nopeasti aikaan ohjelmia, jotka käsittelevät ääntä, midi-signaalia, 3D-grafiikkaa tai vaikka ulkoisten laitteiden kuten peliohjainten ja anturien tuottamaa dataa. Pure Data -ohjelman toimintaa voi muuttaa nopeasti lennossa, minkä vuoksi kieli sopii hyvin luovaan työskentelyyn.

### Mistä sen saa?

Pure Data toimii Windowsissa, OS X:ssä sekä Linuxissa, ja siitä on versioita muillekin laitteille, kuten Iphonelle ja PlayStationille.

Pure Dataa jaetaan kahtena versiona. Pd-vanilla on kielen kehittäjän, Miller S. Pucketten paketoima versio, joka sisältää vain kielen ydintoiminnot. Pure Data -yhteisö puolestaan julkaisee Pd-extended-pakettia, joka sisältää lisäominaisuuksia ja kirjastoja. Molemmat versiot voi ladata Pure Data -yhteisön ylläpitämältä puredata.info-sivustolta. Ohjelma on saatavissa myös monien Linux-käyttöjärjestelmien pakettivarastoista.

Tässä artikkelissa rakennetaan esimerkkiohjelma, joka toimii kaikilla Pure Data -versioilla ja jokaisessa käyttöjärjestelmässä.

### Muutamia perusasioita

Pure Data on graafinen ohjelmointikieli. Siinä ei naputella tekstimuotoista koodia vaan ohjelmat luodaan kytkemällä johtojen avulla erilaisia toimintolaatikoita toisiinsa. Laatikoiden lävitse virtaa dataa, ja ohjelman toimintalogiikka luodaan ohjaamalla ja muokkaamalla datavirtoja.

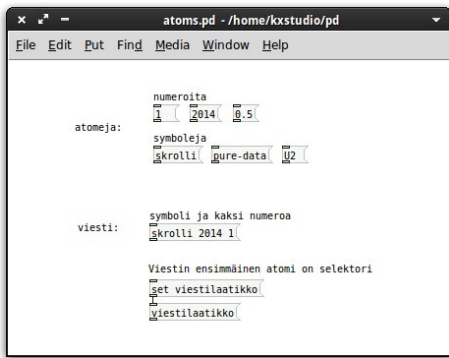
Laatikoita, joista ohjelmat koostuvat, kutsutaan objekteiksi. Niillä kaikilla on

oma nimensä ja tehtävänsä. Objekteilla on tulo- tai lähtöliitäntöjä, joista jokainen välittää tietyn tyyppistä dataa. Datan tyyppin selvittämisessä helpottaa ohjelman Edit-valikon Autotips-ominaisuus. Sen avulla voi tarkistaa, millaista dataa objektin tulot ja lähdöt välittävät.

Objekteja on kahta päätyyppiä: toiset käsittelevät äänisignaalia ja toiset numeroita tai tekstiä. Nämä erottaa toisistaan siitä, että äänisignaalia käsittelevien objektien nimi päättyy tilde-merkkiin (~). Äänisignaali liikkuu paksummissa kaapeleissa kuin muu data.

Muu data koostuu atomeista, jotka ovat välilyönneillä toisistaan erotettuja tekstinpätkiä tai numeroita. Tekstimuotoisia atomeja kutsutaan symboleiksi ja numeromuotoisia atomeja numeroiksi. Symbolissa eli tekstimuotoisessa atomissa voi olla mitä tahansa merkkejä – siis myös numeroita. Numeroatomi sen sijaan koostuu pelkästään numeroista.

Dataa lähetetään johtoja pitkin, jolloin siitä muodostuu viesti. Viesti on yhden tai useamman välilyönneillä erotetun



Kuva 1. Atomeja ja viesteistä.

atomin joukko. Kuvassa 1 on esimerkkejä atomeista ja viesteistä.

Jos viestin ensimmäinen atomi on tekstiä, sitä kutsutaan selektoriksi. Sen avulla objekti tunnistaa sisään tulevan viestin tyyppin. Objektit reagoivat useisiin eri selektoreihin. Kuvassa 1 alempi viesti alkaa atomilla "set". Se on selektori, joka tarkoittaa 'aset viestilaatikon teksti'. Kun ylempää viestiä klikataan, sen lähtöön kytkettyyn toiseen viestilaatikkoon ilmestyy teksti "viestilaatikko".

Bang on hyvin yleinen viesti, jota tarvitaan Pure Datassa usein. Sillä käynnistetään muita objekteja. Lähes kaikki Pure Datan objektit reagoivat bang-viestiin käynnistymällä tai lähettämällä sisältämänsä datan ulos, joten bang-viestin tarkoituksen voisikin ilmaista sanoilla "tee jotain". Bang ei yleensä muuta objektin sisältöä mitenkään.

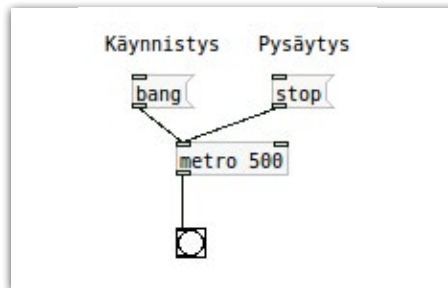
Objekteilla voi olla useita tuloja ja lähtöjä. Näistä ensisijainen on vasemmanpuoleisin tulo. Siihen lähetetty viesti saa yleensä objektin lähettämään dataa eteenpäin lähtöliitäntästään. Muihin tuloihin saapuneet viestit puolestaan tallentuvat objektiin ja jäävät odottamaan viestiä, joka saapuu vasemmanpuoleisimpaan tuloon. Tähän sääntöön on joitakin poikkeuksia, mutta niitä emme käsittele tässä artikkelissa.

## Sekvensserin laatiminen

Tutustumme Pure Datan alkeisiin rakentamalla yksinkertaisen sekvensserin, joka soittaa neljän nuotin melodiaa ja pystyy lähettämään niitä midi-syntetisaattorille. Sekvensserin moottorina on tasaista pulssia lähettävä metro-objekti. Sen perään kytketään neljään laskeva laskuri, jonka avulla iskut ohjataan eri midi-nuotteihin. Nuottien sävelkorkeuden säätämiseksi tehdään yksinkertainen käyttöliittymä ja lopuksi rakennetaan vielä lähtöliitäntä, jonka avulla ohjelman saa kytkettyä midi-syntetisaattoriin.

Rakennetaan aluksi kuvan 2 mukainen metronomi. Pure Data -ohjelmassa uusi dokumentti luodaan valitsemalla

File-valikosta New, jolloin avautuu tyhjä ikkuna työskentelyä varten. Objekteja voi lisätä Put-valikon kautta. Metro-objekti luodaan valitsemalla Put → Object ja kirjoittamalla laatikkoon teksti "metro 500". Metron yläpuolella olevat tuloihin kytketyt laatikot ovat Messageja (Put → Message), ja metron lähtöön kytketty objekti on Bang. Objektit yhdistetään vetämällä hiirellä viiva yhden objektin lähtöliitännästä toisen tuloliitäntään.



Kuva 2. Yksinkertainen metronomi.

Metro-objektin nimen perässä on välilyönti ja numero 500. Tämä on objektille annettu alkuarvo. Alkuarvon voi korvata lähettämällä metron oikeanpuoleiseen tuloon jonkin toisen numeron. Metro-objekti muistuttaa perinteistä metronomia. Sen vasempaan tuloon lähetetään bang-viesti, jonka saatuaan metro käynnistyy ja lähettää bang-viestejä alkuarvona annetun 500 millisekunnin eli puolen sekunnin välein.

Metronomia pääsee testaamaan siirtymällä muokkaustilasta suoritustilaan. Se tapahtuu valikosta Edit → Edit mode. Kun bang-viestiä klikataan, metron perään kytketty graafinen bang alkaa välkyä tasaisesti.

Kun metronomin käynnistämisen jälkeen siirrytään takaisin muokkaustilaan (Edit → Edit mode), huomataan, että bang jatkaa edelleen välkkymistään: Pure Datassa ohjelmat ovat käynnissä kaiken aikaa. Kannattaa siis pitää varansa, sillä muokkauksen seurauksena ohjelmat saattavat mennä erikoisiin virhetiloihin, joihin ei normaalisti jouduttaisi lainkaan. Varma keino ohjelman nollaamiseen on tallentaa tiedosto, sulkea se ja avata uudestaan.

## Laskuri floatin avulla

Nyt kun metro lähettää tasaisesti iskuja, on aika jakaa ne neljän rykelmiin laskurin avulla. Laskureita voi tehdä Pure Datassa monella tavalla, ja nyt tutustumme yhteen melko yksinkertaiseen malliin.

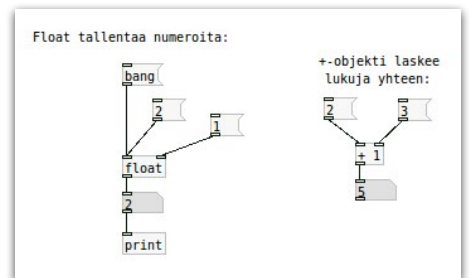
Tarvittavia objekteja on kaksi: float ja +. Float-objekti tallentaa luvun, joka sen tuloliitäntään lähetetään. +-objekti puolestaan laskee tuloliitäntään saapuneet

luvut yhteen.

Floatissa on kaksi tuloliitäntää: vasemmanpuoleinen tallentaa luvun ja saman tien myös lähettää sen ulos. Oikeanpuoleinen tulo ainoastaan tallentaa luvun odottamaan bang-viestiä. Jos haluaa tutkia floatin toimintaa tarkemmin, kannattaa rakentaa kuvan 3 mukainen ohjelma. Float-objektin lähtöön kytketty objekti on nimeltään number (Put → Number). Se näyttää numerot, jotka float on lähettänyt ulos.

Joka kerta kun kuvan 3 bang-viestiä klikataan, float-objekti lähettää eteenpäin siihen tallennetun numeron. Tämän voi todeta print-objektin avulla. Print tulostaa Pure Datan pääikkunaan kaiken datan, joka objektiin on saapunut, joten se on hyvin hyödyllinen ohjelmien testausvaiheessa.

Kuvan 3 ohjelmassa on myös +-objekti, jolla lasketaan yhteen numeroita. Sen tulosten ja lähtöjen toimintaperiaate on täysin sama kuin floatillakin.



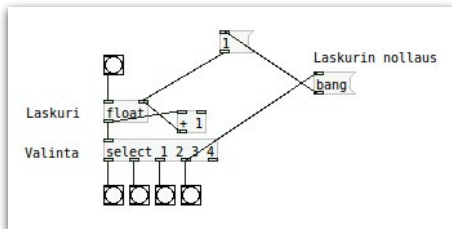
Kuva 3. Float- ja +-objektien toiminnan testausta.

Laskurin luominen on varsin helppoa, ja kuvassa 4 näkyy, miten se tapahtuu. Kun float-objekti saa bang-viestin, se lähettää sisältönsä +-objektille. Tämä kasvattaa lukua yhdellä ja tallentaa tuloksen takaisin float-objektiin oikeanpuoleisen tuloliitäntän kautta.

Uusi objekti tässä ohjelmassa on select. Se ottaa tuloonsa numeroita ja vertaa jokaista saapunutta numeroa alkuarvona annettuihin numeroihin. Tässä tapauksessa olemme antaneet objektille neljä alkuarvoa: 1, 2, 3, 4. Select luo jokaiselle oman lähtönsä ja lähettää bangin siitä lähdöstä, joka vastaa tuloon saapunutta numeroa. Jos sisään tullut numero ei vastaa mitään alkuarvoista, select ohjaa sen ulos viidennestä lähdöstään.

Haluamme, että laskuri palaa aina nelosen jälkeen ykköseen. Se onnistuu lähettämällä neljännessä select-objektin lähdöstä tuleva bang-viesti 1-viestilaatikon kautta, joka puolestaan kytketään laskuriin. Koska viesti kulkee alhaalta takaisin ylöspäin, eli Pure Datassa epätavalliseen suuntaan, tulee ohjelmasta helposti sotkuinen. Kytkentää kannattaa selkeyttää

käyttämällä ylimääräistä bang-viestilaitteikkoa, jonka avulla johdot voi kierrättää muiden objektien ympäri. Näin on tehty kuvan 4 ohjelmassa.



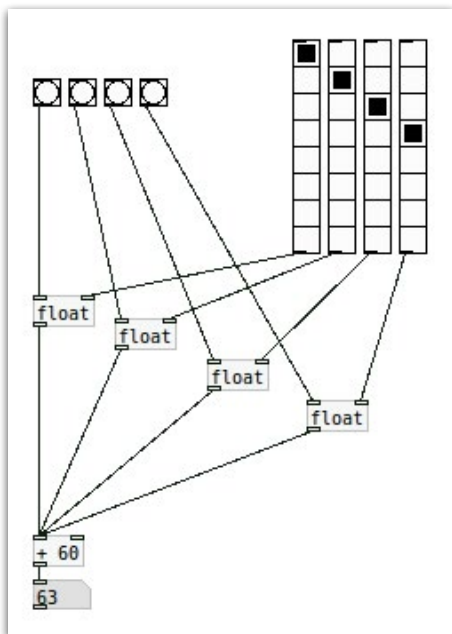
Kuva 4. Laskuri ja valintatoiminto.

## Nuottien luominen ja lähettäminen

Yhdessä metronomi ja laskuri (kuvat 2 ja 4) muodostavat neljään laskevan metronomin, joka lähettää tiedon, missä iskussa ollaan menossa. Sen avulla voidaan rakentaa neljän nuotin mittainen sekvenssi.

Midi-järjestelmässä jokaiselle sävelle on oma numeronsa. Alin sävel on 0, ja numerot kasvavat siitä ylöspäin aina 127:ään asti. Tätä voi ajatella vaikkapa pianon koskettimien kautta. Konserttiflyygelin koskettimisto alkaa midi-nuotti 21:stä, ja sen korkein sävel on midi-arvona 108. Keski-c on 60.

Jos rakennamme järjestelmän, jossa jokainen laskuri-valitsimelta tuleva bang-viesti kulkee yhtä midi-nuottia säilyttävään float-objektiin, saadaan aikaiseksi yksinkertainen neljän nuotin sekvensseri. Kuva 5 havainnollistaa tätä.



Kuva 5. Midi-nuotit ja sävelkorkeuden säätö.

Float-objektin sisältämää midi-nuottia voi muuttaa lähettämällä uuden numeron sen oikeanpuoleiseen tuloon. Tähän tarkoitukseen sopii hyvin vradio-objekti (Put → Vradio), joka luo pystysuuntaisen

painikerivin. Tehdään näitä rivejä neljä: yksi jokaista nuottia varten. Aina kun painikerivin painiketta vaihdetaan, vradio-objekti lähettää painikkeen mukaisen numeron. Numerointi alkaa nollassa, eli esimerkiksi kahdeksan painikkeen ryhmä lähettää numeroita 0–7. Jokaiseen float-objektiin tallentuu siis jokin numero 0–7.

Koska midi-nuotit 0–7 ovat aivan liian matalia järkevän melodian soittamiseen, lähetetään float-objektien sisältö vielä yhteenlaskuobjektin (+) läpi. Sopiva lisättävä luku on esimerkiksi 60, joka tarkoittaa pianon keski-c:tä. Tällöin sekvensseri soittaisi midinuotteja 60–67, mikä vastaa musiikin teoriassa puhtaan kvintin aluetta c:stä g:hen.

Kun tähän mennessä rakentamamme ohjelmat (kuvat 2, 4 ja 5) kytketään yhteen, valmiina pitäisi olla laite, joka lähettää tasaisin väliajoin neljää eri midi-nuotin arvoa. Meidän pitää vielä luoda laitteelle midi-liitäntä, mutta ennen sitä tutustumme Pure Datassa olevaan hyvin kätevään ominaisuuteen: aliohjelmiin ja niiden käyttöliittymiin.

## Sekvensserin siirtäminen aliohjelmiaan

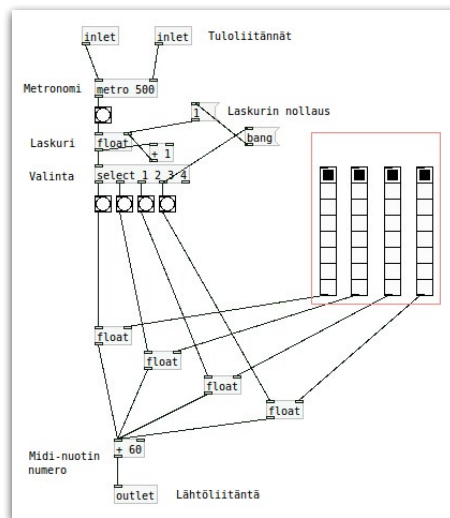
Pure Datassa ohjelmia voi laittaa sisäkän. Tietokoneen levyllä samassa kansiossa olevat Pure Data -ohjelmat näkevät toisensa. Jos esimerkiksi luodaan tiedosto nimeltä pääohjelma.pd ja tallennetaan se samaan kansioon tiedoston sekvensseri.pd kanssa, jälkimmäisen voi sisällyttää ensimmäiseen luomalla objektin (Put → Object) ja kirjoittamalla sen nimeksi "sekvensseri".

Aliohjelmat näkyvät pääohjelmassa omina objekteina. Kun niitä klikkaa hiiren oikeanpuoleisella napilla ja valitsee avautuvasta valikosta Open, aliohjelma aukeaa omaan ikkunaan, jossa sitä voi muokata. Pure Datan aliohjelmat ovat ihan samanlaisia objekteja kuin kielen valmiit objektitkin. Niillä voi olla tulo- ja lähtöliitäntöjä sekä oma käyttöliittymä.

Siirrytään teoriasta käytäntöön. Yhdistämme aiemmin luodut ohjelmat samaan ja teemme niistä aliohjelman nimeltä sekvensseri.pd (kuva 6). Luomme aliohjelmalle kaksi tuloliitäntää ja yhden lähtöliitäntän, joiden kautta pääohjelma voi käyttää sitä. Liitännät luodaan inlet- ja outlet-objekteilla.

Pure Datan hienouksiin kuuluu, että aliohjelmille voi tehdä käyttöliittymän, joka näkyy pääohjelmassa. Aliohjelmaikkunaan luodaan sitä varten käyttöliittymäalue.

Ominaisuus on nimeltään "Graph on parent", ja sen saa käyttöön klikkaamalla



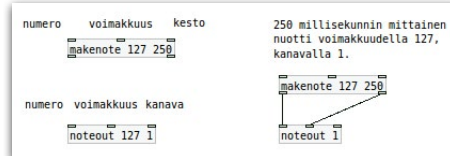
Kuva 6. Valmis sekvensseri.pd-aliohjelma.

la hiiren oikeanpuoleisella napilla aliohjelmaikkunan tyhjää valkoista aluetta ja valitsemalla Properties. Avautuu ikkuna, josta ominaisuuden saa päälle. Ikkunassa on kentät, joiden avulla voi määrittää käyttöliittymäalueen sijainnin ja koon.

Aluetta merkitsemään ilmestyy punainen suorakaide, jonka sisälle asetetut elementit näkyvät aliohjelmasta ulospäin. Siirretään alueelle nuottien sävelkorkeutta säätävät painikkeet. Valmis aliohjelma ja sen punainen käyttöliittymäalue näkyvät kuvassa 6.

Aliohjelma sekvensseri.pd kannattaa tallentaa Pure Datan omaan kirjastokansioon, jolloin se on automaattisesti aina käytettävissä. Kansio sijaitsee eri käyttöjärjestelmissä eri paikassa: Windows-versiossa pd-kansion sisällä on extra-kansio. Linuxissa jokaisen käyttäjän kotikansiossa on pd-externals-niminen kansio.

Nyt pääohjelmassa voidaan käyttää objektia "sekvensseri", jossa näkyvät painikerivit. Objektilla on kaksi tuloa, joista toinen on bang- ja stop-viesteille ja toinen säätää sekvensserin lyöntitiheyttä. Objektin alareunassa on lähtö, joka antaa midi-nuottien numeroita. Sen otamme käyttöön seuraavaksi.



Kuva 7. Midi-lähdön rakentaminen.

## Midin lähettäminen

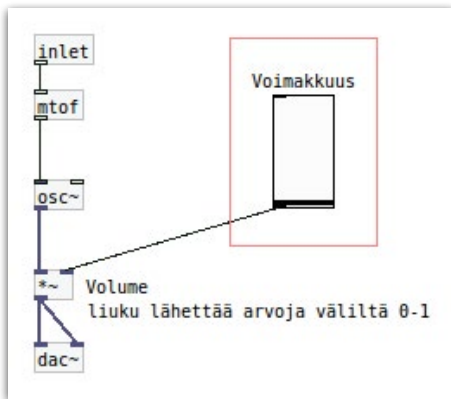
Pure Datassa on kaksi objektia, joiden avulla on helppoa lähettää midi-nuotteja: noteout ja makenote. Noteout on näistä yksinkertaisempi. Sille annetaan nuotin numero, soittovoimakkuus ja midi-kanava. Noteoutille pitää myös lähettää erillinen viesti, kun haluaa sammuttaa nuotin.



Makenote helpottaa tätä, sillä se luo nuotille automaattisesti sammutuskäskyn.

Kuvassa 7 esitellään näiden objektien tuloliitäntöjen tarkoitukset ja näytetään, miten ne kytketään yhteen. Kun sekvensserin lähtö yhdistetään kuvassa näkyvän kytkennän makenote-objektin vasempaan tuloon, pitäisi ohjelman lähettää midi-nuotteja kanavalla yksi.

Pure Datassa midi ei kuitenkaan ole oletuksena päällä, joten on vielä mentävä Media-valikkoon käynnistämään se. Valikon tarjoamat vaihtoehdot ovat eri käyttöjärjestelmissä hieman erilaisia. Windowsissa ja OS X:ssä mennään Midi settings -valikkoon ja valitaan jokin järjestelmän tarjoamista midi-laitteista. Linuxissa luodaan Jack midi- tai Alsa midi-portti, jonka kautta Pure Datan voi yhdistää muihin midiä käyttäviin ohjelmiin. Väliin tarvitaan vielä reititysohjelma, esimerkiksi Qjackctl tai Kxstudion mukana tuleva Claudia.



Kuva 8. Oskillaattori ja äänenvoimakkuuden säädin.

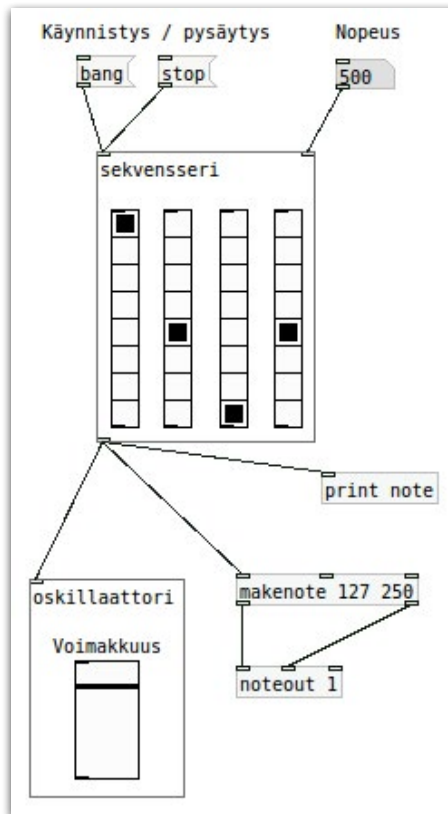
### Yksinkertainen oskillaattori

Mikäli sinulla ei ole yhtään midillä ohjattavaa syntetisaattoria, voit rakentaa yksinkertaisen oskillaattorin testatakseen sekvensseriä. Esimerkkioskillaattori (kuva 8) ottaa sisäänsä nuotin midi-numeron ja muuntaa sen oikealla taajuudella soivaksi siniaalloksi.

Oskillaattorissa oleva mtof-objekti muuntaa midi-nuotit taajuuslukemaksi (Hz) ja osc~-objekti lähettää taajuuslukeman mukaista siniaaltoa. \*~-objektin avulla toteutetaan äänenvoimakkuuden säätö. Sisään tuleva siniaalto kerrotaan desimaaliluvulla 0-1, joka saadaan liukusäätimeltä. Liukusäädin on vslider-objekti, jonka asetuksia pääsee muokkaamaan klikkaamalla objektia hiiren oikeanpuoleisella napilla ja valitsemalla Properties. Avautuu ikkuna, josta voi valita säätimelle muun muassa otsikon (label) sekä lähtöarvon alarajan (bottom) ja ylärajan (top). Asetetaan alarajaksi 0 ja ylärajaksi 1.

Äänisignaalin tuottamiseen tarvitaan vielä dac~-objekti, jolle syötetään osc~:n ja \*~:n tuottama siniaalto. Oskillaattorista ei heti kuulu ääntä, sillä Pure Datassa pitää erikseen kytkeä äänisignaalin käsittely eli DSP päälle. Tämä onnistuu Pure Datan pääikkunassa klikkaamalla rasti DSP-tekstin vieressä olevaan ruutuun.

Kun kuvan 8 mukainen ohjelma tallennetaan samaan kansioon aiemmin luotujen ohjelmien kanssa, sitäkin voi käyttää aliohjelmana. Sopiva nimi on esimerkiksi oskillaattori.pd. Kun oskillaattori kytketään sekvensserin lähtöön, tietokoneen kaiuttimista pitäisi kuulua sekvensserin melodia siniaallon soittamana.



Kuva 9. Valmis ohjelma, jossa sekvensseri ohjaa sekä oskillaattoria että midi-laitetta.

### Valmista tuli

Valmis ohjelma näyttää lopulta hyvin yksinkertaiselta (kuva 9). Ohjelma käynnistetään ja pysäytetään yläreunassa näkyvillä bang- ja stop-viesteillä. Sekvensseri ja oskillaattori ovat aliohjelmaa ja näkyvät pääohjelmassa omina objekteinaan. Sekvensseri on kytketty ohjaamaan sekä oskillaattoria että midi-laitetta. Ääni saattaa siis kuulua kahdesta äänilähteestä. Ohjelmassa oleva print-objekti tulostaa midi-nuottien numerot Pure Data -ohjelman pääikkunaan.

Tässä artikkelissa on vasta raapaistu Pure Datan pintaa. Kieli kykenee käsittelemään ääntä, kuvaa tai vaikka tekstidostoja monipuolisesti. Sillä voi esimerkiksi tehdä interaktiivisia sävellyksiä tai

omia työkaluja midi-signaalin käsittelemiseen. Toivottavasti esimerkki osoitti, että Pure Datan avulla käyttökelpoisten ohjelmien tekeminen ei vie kovin paljon aikaa eikä ole vaikeatakaan. 🐼

### Lisätietoa

Pure Data -ohjelman Help-valikossa on toimintoja, joiden avulla saa lisätietoa ohjelmasta ja ohjelmointikielystä. Myös jokaisella objektilla on oma ohjeensa, jonka saa avattua klikkaamalla objektin päällä hiiren oikeanpuoleisella napilla ja valitsemalla Help. Lisäksi internetissä on muutamia hyviä oppaita ja sivustoja. Myös yhteisön apua on saatavilla. Pure Data -yhteisö kokoontuu esimerkiksi ircissä, sähköpostilistalla ja keskusteluforumilla.

Ohjekirja: <http://flossmanuals.net/pure-data/>  
Tietoa yhteisöstä: <http://puredata.info/community>  
Keskustelufoorumi: <http://puredata.hurlleur.com>

### Ideita ohjelman laajentamiseen

- Neljän nuotin melodia on hieno, mutta kahdeksan nuotin melodia olisi vielä hienempi. Laajenna sekvensseriä kahdeksan nuotin mittaiseksi ja lisää sävelkorkeuksia kahdeksasta kahteentoista.
- Tee sekvensseriin toiminto, jolla melodian sävelkorkeutta voi muuttaa oktaavilla ylös tai alas. Oktaavi on 12 säveltä.
- Luo ohjelmaan midi-kanavan vaihtotoiminto. Kokeile esimerkiksi number2-objektia. Sitten on helppoa luoda useita sekvenssereitä ja laittaa ne ohjaamaan eri syntetisaattoreita.
- Sekvensserin lyöntitiheyttä voi säätää sekunnin tuhannesosissa kytkemällä sen oikeanpuoleiseen tuloon liukusäätimen tai numero-laikikon. Musiikissa on kuitenkin tapana käyttää tempon mittana bpm-lukua eli iskuja minuutissa. Tee sekvensseriin ominaisuus, joka muuntaa bpm-lukeman tuhannesosasekunneiksi metroa varten.
- Lisää oskillaattoriin vireen säädin: mtof-objektin lähettämään taajuuslukuun voi +-objektin avulla lisätä esimerkiksi liukusäätimen tuottamia lukuja.
- Opetta sekvensseri lukemaan syötettä joltakin tietokoneeseen kytketyltä laitteelta ja muuttamaan sävelkorkeuksia sillä perusteella. Pure Datassa on omat objektinsa niin näppäimistön (key, keyup, keyname), hiiren (cursor) kuin peliohjaintenkin (hid, Window-sissa hidin) lukemiseen. Midi-koskettimistöjen lähettämää signaalia voi analysoida objekteilla notein, ctlin ja midiin.
- Ohjelman laajentamisen kannalta hyödyllisiä objekteja ovat esimerkiksi send, receive, loadbang, tgl ja cnv.



# 60 vuotta suomalaisia tietokoneita

*Vaikka suomalaiset ovatkin käyttäneet enimmäkseen ulkomaisia tietokoneita, on Suomella ollut myös omaa tuotantoa jo 60 vuoden ajan.*

Teksti: Ville-Matias Heikkilä Kuvat: Mikko Rasa, Jari Viitala, Tekniikan museo, Wikimedia Commons

## Epäonninen Esko

Monien muiden maiden tapaan myös Suomessa oli jonkinlaisia tietojenkäsittelylaitteita jo ennen varsinaisia tietokoneita. Fysiikan ongelmien ratkaisemiseen rakennettiin TKK:lla analogiakoneita, ja isoilla virastoilla oli käytössään reikäkorttikoneita.

Ruotsalaiset olivat rakentaneet 1950-luvun alkupuolella jo kaksi yleiskäyttöistä tietokonetta: relepohjaisen Barkin (1950) ja elektroniputkiin perustuvan Beskin (1953). Suomalaisenkin piti siis saada maahan oma ”matematiikkakone”. Hanketta varten perustettiin vuonna 1954 Matematiikkakonekomitea, jonka vetäjäksi tuli Helsingin yliopiston matematiikan professori Rolf Nevanlinna.

Suomi oli köyhä maa ja budjetit rajalliset, joten täysin omaperäisen laitteen sijaan päätettiin rakentaa kopio ulkomaisesta koneesta. Komitea kävi tämän vuoksi Ruotsissa, Saksassa ja Sveitsissä tutustumassa sikäläisten aikaansaannoksiin.

Ruotsin Besk ihastutti suomalaisia vierailijoita. Amerikkalaisen John von Neumannin ideoihin perustuneen laitteen ohjelmointiominaisuuksia pidettiin loistavina, ja se oli ollut hetken aikaa jopa maailman nopein tietokone: 20 000

yhteenlaskua sekunnissa. Se arvioitiin kuitenkin turhan kalliiksi kokonsa vuoksi: elektroniputkia olisi tarvittu noin 2 400 kappaletta.

Saksan Göttingenin yliopistossa komitea tutustui rakenteilla olevaan koneeseen nimeltä G1a, josta piti tulla putkipohjainen pikkuveli vuonna 1952 valmistuneelle G1-relekoneelle. Minimalistisen koneen rakentamiseen riittäisi noin 500 putkea, mikä vetosi suomalaisiin. Vaihtoehtoja harkittuaan Matematiikkakonekomitea päätyi ostamaan G1a:n piirustukset.

Koneelle annettiin nimeksi Esko, Elektroninen sarjakomputaattori, ja siltä odotettiin paljon. Suomen Kuvalehti hehkutti sitä peräti uudeksi Sammoksi, ja koneen uskottiin tyydyttävän kaikki Suomen laskentatarpeet vuosiksi eteenpäin.

Suomalaiset eivät kuitenkaan tienneet, että keskeneräisen koneen piirustuksetkin olivat keskeneräiset. Monet paperilla toimineet insinöörikikkailut eivät toimineet käytännössä, ja putkitekniikan viritäminen äärimmilleen teki siitä epäluotettavaa. Saksalaiset riitautuivat keskenään ja siirsivät resursseja G1a:lta G2- ja G3-koneiden kehitykseen. Lopulta suomalaiset joutuivat itse suunnittelemaan konetta uusiksi ja jopa avustamaan saksalaisia. Aikataulut ja budjetit

venyivät. Saksalaiset saivat ensimmäisen G1a:n valmiiksi 1958, ja Suomen Esko valmistui 1960.

Eskolle ei ollut valmistuessaan enää paljon käyttöä, koska se oli hidas ja vanhanaikainen. Se siirrettiin eläkkeelle yliopiston laskentakeskuksesta jo pari vuotta valmistumisensa jälkeen. Hankkeeseen oli tuhlaantunut rahaa paljon enemmän kuin Beskin kopioimiseen olisi mennyt. Postisäästöpankki ei ollut malttanut odottaa Eskon valmistumista vaan oli hankkinut vuonna 1958 Yhdysvalloista IBM 650:n, joka sai nimen Ensi.

G1a:n ja Eskon juuret olivat 1940-luvun saksalaistekniikassa. Käskyjä ei tallennettu muistiin, vaan ne suoritettiin suoraan reikänauhalta kuten Konrad Zusen varhaisissa relelaskimissa. Optisia reikänauhanlukijoita oli koneessa kaikkiaan kymmenen, ja aliohjelmille ja silmuikoille varatut nauhat piti teipata renkaiksi, jotta toisto onnistuisi. Keskusmuistina toimi 1 840 sanan rumpumuisti. Konservatiivinen tekniikka yhdistettynä minimalistiseen nuukailuun teki Eskosta varsin hitaan: se pystyi vain pariinkymmeneen yhteenlaskuun sekunnissa. Koneen sananleveys oli kuitenkin peräti 60 bittiä, ja kaikki laskenta suoritettiin liukuluvuilla.

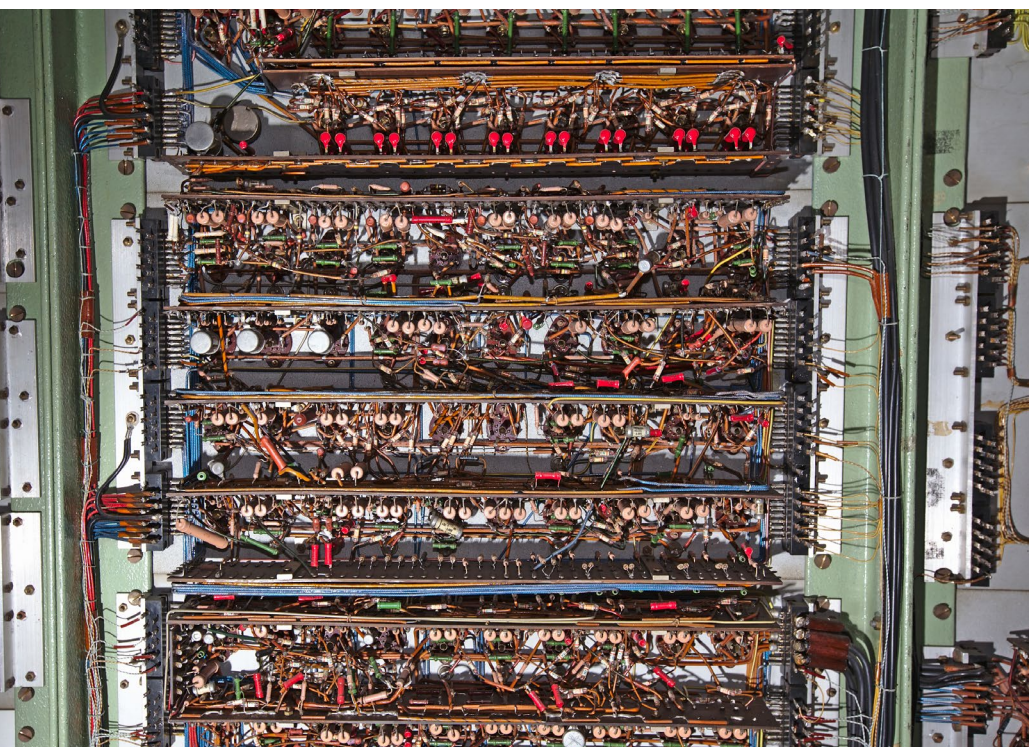
Eskon sivutuotteena syntyi myös ensimmäinen suomalainen digitaalinen



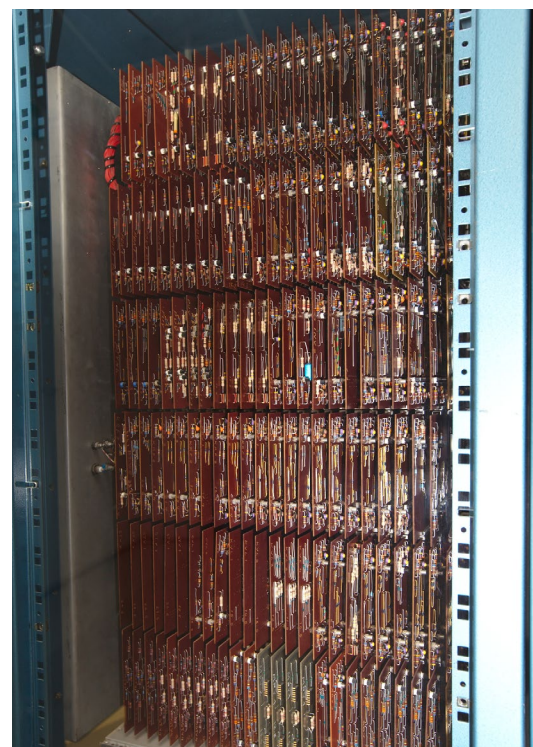
Esko-tietokoneen putkia. Vasemmalla sivulla ohjauspaneeli.



Reflacin hallintapaneeli.



Eskon virtapiirejä.



Reflacin virtapiirejä.

pelii, Nim-peliä pelaava automaatti. Automaatin rakensi vuonna 1955 Matematiikkakonekomiteaan kuulunut Hans Andersin omalla harjoitusprojektinaan.

### Ei tietokonetehtästä vielä 1960-luvulla

Monet maat aloittivat tietokoneiden sarjatuotannon ensimmäisten onnistuneiden rakennusprojektien myötä. Ruotsiin syntyi Datasaab, Norjaan Norsk Data, Tanskaan Regnecentralen, ja toki myös Neuvostoliittoon nousi tietokoneita. Suomen Esko ei kuitenkaan tarjonnut hyvää pohjaa omalle tuotannolle, eikä asiaan tohdittu muutenkaan panostaa epäonnisen hankkeen jälkeen. IBM alkoi vallata Suomen markkinoita.

Suomen Kaapelitehdas Oy perusti

vuonna 1960 elektroniikkaosaston, joka oli jonkin aikaa myös IBM:n tärkein suomalaishankkija. Osasto keskittyi aluksi hankkimaan ulkomaisia tietokoneita ja myymään niitä eri tahoille. Voitoilla rahoitettiin oman elektroniikkatuotannon kehitystä siten, että jossain vaiheessa voisi aloittaa tietokoneutuotannon. Vuonna 1967 Kaapelitehdas fuusioitui Nokiaan ja elektroniikkaosaston nimeksi tuli Nokia Elektroniikka.

Ensimmäinen täysin Suomessa suunniteltu ja rakennettu tietokone oli TTK:n Reflac (Reflex Arithmetics Computer). Vuonna 1967 valmistuneen koneen suunnitteli Matti Kilpi, ja hankkeen vetäjä oli myöhemmistä neuroverkkotutkimuksistaan tunnettu Teuvo Kohonen. Innovatiivisena pidetyn koneen logiikka oli

transistoripohjaista, ja ohjelma ajettiin ferriittirengasmuistilla. Kone oli tarkoitettu teollisuuden prosessinohjauksen tutkimukseen.

Suomen ensimmäinen sarjana tuotettu tietokone oli sähköyhtiö Strömbergin Selco-1000, jota valmistettiin vuodesta 1970 alkaen yhdeksän kappaletta. Kone perustui osittain Reflacin ja oli niin ikään tarkoitettu teollisuuskäyttöön. Useimmat Selcot asennettiin paperitehtaisiin. Kone oli teknisesti hyvin tarkoituksenmukainen monipuolisine I/O- ja keskeytysjärjestelmineen, mutta se jäi massatuotettujen ulkomaisten minitietokoneiden kuten PDP-8:n ja PDP-11:n jalkoihin. Selco oli Reflacin tapaan suunniteltultaan kotimainen, aina käskykanta ja ohjelmointikieliä myöten.



Nokia Mikko 3.

### Minitietokoneiden aika

Nokia alkoi valmistaa omia tietokoneitaan 1970-luvun alkupuolella. Konesarjan nimeksi tuli Mikko, joka on suomalistettu lyhennelmä sanoista "micro computer". Tämä ei kuitenkaan tarkoittanut mikrotietokonetta sanan myöhemmin vakiintuneessa merkityksessä vaan vähän tavallista pienempää minitietokonetta.

Mikko 1:tä (1972) käytettiin enimmäkseen teollisuuden tiedonkeruussa mutta myös Alkon kassapäätteissä. Mikko 2 (1973) puolestaan kehitettiin pankkeja varten. Suurin asiakas oli Kansallis-Osake-Pankki, joka halusi Mikko 2:n ja siihen liittyvät kassapäätteet jokaiseen konttoriinsa. Vuosikymmenen lopulla ilmestyi vielä Mikko 3, jonka I/O-toiminnot oli uusittu. Mikoissa pyrittiin mahdollisimman vähäiseen resurssien käyttöön, ja muun muassa keskus koneen 24 kilotavun keskusmuisti riitti pankkikonttorin kaikkiin tarpeisiin. Ohjelmistosta vastasi Nokian tytäryhtiö Softplan.

Myöhemmin Nokian minitietokonesarjaan liittyi vielä superminikone MPS-10 (1983), joka oli rakennettu Ada-ohjelmointikielen ympärille. Ada on Yhdysvaltain armeijan kehittämä kieli, joka tunnetaan erityisen vahvoista luotettavuus- ja tietoturvatarkistuksistaan.

Muista suomalaisista minitietokoneista mainittakoon valtion tietoliikenneyhtiö Televan TDS-8, jolle Teko-tekstinkäsittelyohjelman ensimmäinen versio tehtiin. Televa kehitti Nokian kanssa myös digitaalista puhelinkeskusjärjestelmää, joka aloitti DX200-tuoteperehen vuonna 1978.

Suomalaiset tietotekniikkayritykset

ja valtion virastot halusivat myös yhdistää voimansa. Suomalaisen tietokoneprojektin eli Sutin tarkoituksena oli aloittaa mikroprosessorien, minitietokoneiden ja ohjelmistojen tuotanto jonkinlaisena kansallisena suurtehtävänä. Suurin osa tuotannosta olisi mennyt vientiin, etenkin sosialistisiin maihin. Yritykset rupesivat kuitenkin vetäytymään projektista, kun 1970-luvun lopulla alkoi näyttää siltä, että syntyvä valtionyhtiö tulisi jyräämään ne alleen.

### Mitä tehdä mikroprosessorilla?

Suomalaisen mikrotietokoneen isänä voidaan pitää Erkki Kurenniemeä, joka tunnetaan paremmin elektronisen musikin pioneerina. Kurenniemi oli käyttänyt digitaalitekniikkaa jo 1960-luvun DIMI-syntetisaattoreissaan ja oli hyvin kiinnostunut myös tietokoneiden mahdollisuuksista. Kurenniemi halusikin saada Intelin keksimän mikroprosessorin käyttöönsä mahdollisimman pian.

Vuoden 1973 alkupuolella Suomeen saatiin kaksi Intelin 8008-suorittinta: toinen meni Nokian tutkimuskeskukselle ja toinen Kurenniemen yritykselle, Digelius Elektronikka Oy:lle. Digelius kehitti suorittimen ympärille yksinkertaisen tietokoneprototyypin, jota käytettiin vilkkuvaloihin ja kytkimiin perustuvalta konsolilta.

Digeliuksen mikrotietokone oli nimeltään DIS System. Digeliuksen kanssa toimi yhteistyössä myös muita elektronikkayrityksiä, jotka kehittivät omia samantapaisia koneitaan. Sähköliikkeiden Oy:n kone oli nimeltään ADT-8 ja perustui melko suoraan DIS:ään. Euroka Oy:n 6800-pohjainen kone oli nimeltään Euro 6. Koneita ei tarkoitettu yleiskäyttöisiksi, vaan niiden päälle rakennettiin erilaisia teollisuussovelluksia, joita myytiin jopa Singaporeen asti. Euro 6:sta kehitettiin 1970-luvun lopulla myös näyttöpäättekäyttöön tarkoitettu pöytämikro, Euro-term 6.

### Telmac ja harrastemikrojen synty

Suomalainen harrastetietokone syntyi vuonna 1977, kun RCA:n mikropiirejä maahan tuonut Telercas toi markkinoille 1802-suorittimeen perustuvan Telmac 1800:n. Konetta myytiin rakennussarjana 2 000 kappaletta, noin puolet Suomeen ja loput Ruotsiin.

Telmac 1800 muistuttaa RCA:n piireihin perustuvia Cosmac ELF/VIP -koneita, jotka julkaistiin rakennusohjeina amerikkalaisessa Popular Electronics -lehdessä. Peruslaitteessa on kaksi kilotavua RAM-muistia, kasettinauhuri- ja televisioli-



Nokia Mikko 3 -koneen näyttö ja näppäimistö.

tännät sekä heksadesimaalinäppäimistö. Videopiiri 1861 on hyvin pelkistetty ja näyttää vain 64 mustavalkopikseliä vaakajuovalla. Tyypillinen resoluutio Telmac-ohjelmissa on 64×32 pikseliä, jota myös Chip-8-ohjelmat käyttävät.

Konetta ohjelmoitiin sekä suoraan 1802-konekielellä että Chip-8-virtuaalikoneen kautta. Chip-8 on käskykannaltaan helpommin lähestyttävä kuin suora konekieli ja sisältää valmiit käskyt grafiikan piirtämiseen. Sille onkin tehty kymmenittäin pelejä. Chip-8-tulkkeja on saatavilla RCA-pohjaisten koneiden lisäksi muillekin alustoille, esimerkiksi graafisille laskimille.

Telmacin suunnittelija Osmo Kainulainen julkaisi Oscom-yrityksensä nimissä koneelle myöhemmin laajennusmoduulin, joka antoi sille aidon tekstiilan ja mahdollisti Tiny Basic -tulkin ajamisen. Koneesta ilmestyi myöhemmin myös parannettuja versioita: Oscom 1000, Telmac 2000 ja edullinen Oscom Nano. Telmac-harrastajilla oli kerho 1800 Users' Club ry, joka julkaisi myös omaa Tieturilehteä.

Telmacin aikaan oli myös pienempiä kotimikroprojekteja. TTK:n elektronikkakerho Juotos tilasi Z80-pohjaisen koneen komponentit yhdessä, ja kukin kolvasi oman koneensa kasaan samoista piirustuksista. Juotoksen koneisiin liitettiin jopa levykeasemia ja niille tehtiin myös oma, hyvin nopeaksi ja tiiviiksi opintoitu Basic-tulkki.

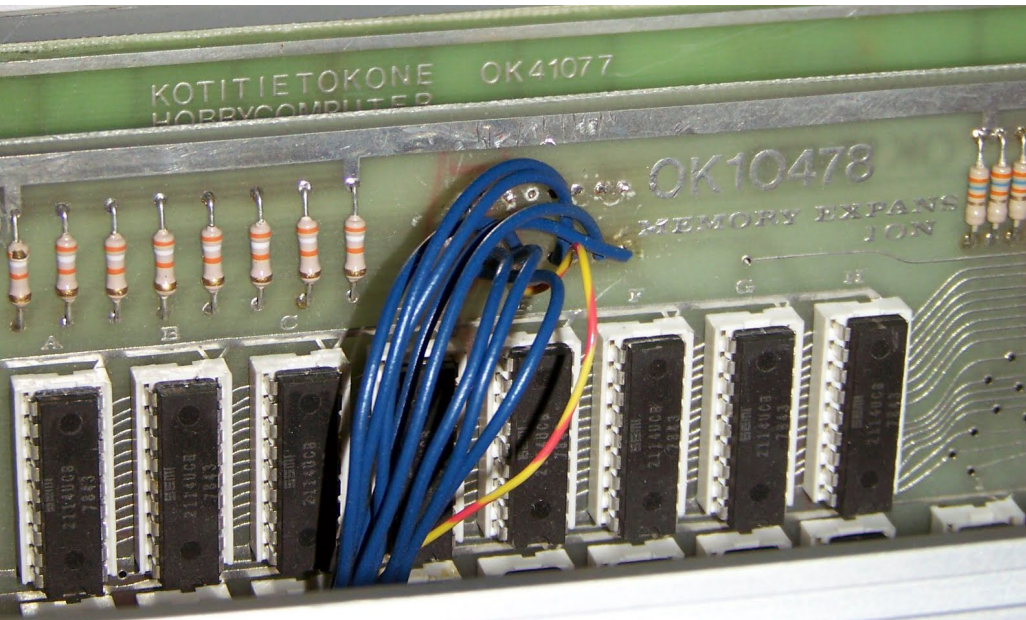
### 80-luvun mikrotietokoneet

Valmiiksi kootut kotimikrot alkoivat yleistyä Suomessa 1980-luvun alussa. Suosituimmat koneet olivat ulkomaisia, mutta myös muutama suomalaisvalmistaja yritti päästä mukaan.

Telercas toi vuonna 1982 markkinoille Telmac TMC-600 -kotimikron. Koneen perusmallissa on yhdeksän kilotavua RAM-muistia, Basic-tulkki, värillinen 40×24 merkin tekstiila (joka matkii myös 80×72 pikselin grafiikkatila) ja



Laajennettu ja koteloitu Telmac 1800.



Telmac 1800:n virtapiirejä.



Telmac TMC-600.

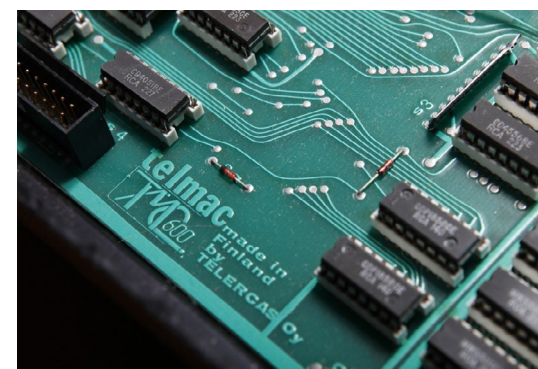
yksi äänikanava. Konetta myytiin kuitenkin vain 600 kappaletta. Vielä tuntemattomampi oli Paavo-Paimen Oy:n PP-200, jonka suorittimena oli Telmacin tapaan RCA 1802.

Paremmiin menestyivät suomalaiset CP/M-koneet, joita tuotettiin toimisto- ja koulukäyttöön. Suomen suosituin alkuaikojen koulutietokone oli kielistudioita valmistaneen Auditekin AMC-100, jota myytiin ulkomaillekin. AMC-100:t pystyi myös kytkemään lähiverkoksi. Toinen koulukäyttöön tarkoitettu CP/M-kone oli Pentti Hakalan suunnittelema Spectra eli Mikrospectra. Myös harrastajat rakentelivat omia CP/M-koneitaan.

Tunnetuin suomalainen CP/M-kone on kuitenkin Nokian Mikromikko 1, josta julkaistiin useita alamalleja vuosina 1981–1984. Mikromikossa on 64 kilotavun keskusmuisti ja 800×327 pikselin tarkkuusgrafiikkatila. Tarkkuusgrafiikka näkyy erikoisesti omalla itsenäisenä tasanaan 80×25-merkkisen tekstitilan taustalla.

Myöhemmät Mikromikot ovat jo x86-koneita. Mikromikko 2 perustuu 80186-suorittimeen mutta ei ole muutoin IBM PC -laiteyhteensopiva. Käyttöjärjestelmänä on Nokian oma versio MS-DOSista, ja BIOSin tilalla on RMX-niminen varusohjelmisto. Mikromikko 3:t puolestaan ovat jo täysin IBM-yhteensopivia 286- ja 386-koneita. Mikromikkoja valmistanut Nokia Data myytiin vuonna 1991 ICL:lle, joka jatkoi tuotemerkin käyttämistä Suomeen myymissään koneissa.

Monille tulee suomalaismikroista puhuttaessa mieleen myös Salora, joka myi 1980-luvun alkupuolella Fellow- ja Manager-nimisiä koneita. Nämä olivat kuitenkin hongkongilaisen Video Technology -yhtiön tekemiä, Saloralle brändätyjä tuotteita. Z80-pohjainen Salora Fellow on sama kuin VTech Laser 200, ja 6502-pohjainen Manager sama kuin Laser 2001. Uudelleenbrändäystä harrasti myös Finlux, joka liimasi myymiinsä Dragon 32 -koneisiin Finlux Dragon -tarrat.



## Nokian suurvaltakausi

Kaapelitehdas ja Salora kehittivät ensimmäiset radiopuhelimensa vuonna 1964. Vuonna 1979 voimat yhdistettiin Mobira Oy:ksi, joka siirtyi kokonaan Nokialle vuonna 1984. Kun digitaalinen GSM-verkko otettiin käyttöön vuonna 1991, alkoivat tietokoneet ja matkapuhelimet lähentyä toisiaan aivan uudella tavalla. Nokia perustikin tämän vuoksi Tampereelle oman mobiilidatalaitteiden kehitysyksikön vuonna 1993.

Vuonna 1996 Tampereella valmistui laite, jota monet pitävät maailman ensimmäisenä onnistuneena älypuhelimena. Nokia Communicator 9000 yhdisti GSM-puhelimen ja qwerty-näppäimistöllisen tasku-pc:n samojen kuorien sisään. Ensimmäisessä Kommunikaattorissa on 386-suoritin, joka ajaa graafista GEOS-käyttöliittymää MS-DOS-yhteensopivan ROM-DOSin päällä. Mustavalkoisen näytön resoluutio on 640×200 pikseliä. Laite oli menestys niin bisnesmiesten kuin kansanedustajienkin keskuudessa. Vuosituhannen vaihduttua Kommunikaattoreissa siirryttiin ARM-suorittimiin ja Symbian-käyttöjärjestelmään. Viimeinen tätä nimeä kantanut laite oli vuonna 2007 ilmestynyt E90 Communicator.

Ennen viihde-elektronikkaosastonsa myyntiä 1990-luvun lopulla Nokia ehti kehittää kaksi hakkerien suosioon päätyntä digiboksia, DBox ja DBox2. Laitteita myytiin virallisesti vain Saksassa ja Itävallassa, mutta Linux-pohjaisen Tuxbox-ohjelmiston avulla niitä pystyy käyttämään muuallakin. DBoxit ovat edelleen suosittuja harrastajapiireissä Tuxboxin tarjoamien piraattiominaisuuksien vuoksi.

Vuonna 2005 tuli myyntiin Noki-

an ensimmäinen tablettitietokone. Nokia 770 perustui yhtiön omaan Linux-pohjaiseen Maemo-käyttöjärjestelmään, jota oli tarkoitus ruveta käyttämään myös kosketusnäyttöpuhelimissa. Pomot tekivät kuitenkin

kohtalokkaan päätöksen haudata Linux-alustat ja keskittyä Symbianiin. Maemo-pohjainen N900 ja Meego-pohjainen N9 jäivät täten Nokian ainoiksi Linux-puhelimiksi. Meegon perintöä jatkaa Nokiasta irtaantunut Jolla Oy.

Nokia kokeili vuonna 2009 jopa paluuta taskukokoa isompiin pc:ihin. Booklet 3G-miniläppäri jäi kuitenkin sarsansa ainoaksi.

Vaikka Nokian matkapuhelintuotanto myytiin Microsoftille vuonna 2013, Nokian tytäryhtiö NSN valmistaa edelleen tietotekniikkaa verkkoyhteyksien ja muiden tietoliikennelaitteiden muodossa.

## Onko meillä tulevaisuutta?

Suomessa on kehitetty paljon hyvää ja omaperäistä tietotekniikkaa, mutta suuri osa siitä on jäänyt ulkomaisen suurtuotannon jalkoihin. Reflac ja Selco olivat aikanaan edistyksellisiä koneita, mutta nykyisin harva on kuullut niistä. Suomea ei voi kuitenkaan pitää epäonnisena: useimpien muiden pienten maiden omat koneet jäivät aivan samalla tavoin marginaaliin, ja olihan meillä sitä paitsi Nokia.

Suomalaiset tietokoneet olivat etenkin alkuvuosikymmeninä pieniä, säästeliäitä ja tarkoituksenmukaisia. Meillä ei



Nokia 9110 Communicator.

rakennettu yhtään omaa suurtietokoneetta vaan ne ostettiin mieluummin muilta. Ei siis ihmeekään, että nimenomaan taskutietotekniikka räjäytti lopulta potin.

Nykyinen tietokonemaailma on aivan erilainen kuin

ennen kännykkähuumaa. Valtaosa mistä tahansa kaupallisesta tuotteesta rakentuu eri puolilla maailmaa tehdyille pohjatyölle. Vaikka Suomessa edelleenkin kehitetään ja valmistetaan sulautettuja järjestelmiä ja muita tietokoneiksi laskettavia laitteita, kuinka suurelta osin niitä voi lopulta sanoa suomalaisiksi?

Puhtaalta pöydältä suunnitellut tai edes suurelta osin omaperäiset tietokoneet ovat yhä useammin harrasteprojekteja. Tee-se-itse-asette on enää harvoin kaupallisesti ja teollisesti järkevää, mutta sitä kannattaa vaalia, jotta kosketus tekniikan perusteisiin säilyy.

Hakkerihenkeä on esimerkiksi Arduinon kaltaisissa avoimen lähdekoodin rakentelualustoissa, joita on viime vuosina kehitetty Suomessakin. Muiku Oy:n vuonna 2012 lanseeraama Aery32 on yleiskäyttöisempi kortti, kun taas prototyyppiasteella oleva ELL-i on tarkoitettu Ethernet-verkosta virtansa saavien pikulaiteiden pohjaksi.

Tulevaisuudessa kehitys saattaa hyvinkin johtaa siihen, että pienimuotoista ja paikallista ruvetaan taas arvostamaan eikä kaikkea haluta enää rakentaa mustien monoliittien varaan. Suomalaisen tietotekniikan uutta nousua odottavien kannattaakin pitää silmällä, mitä esimerkiksi hacklabeissa tapahtuu. 🐛



Nokia Mikromikko 1.



## Politiikan kuuma kevät Tekijänoikeus ja tietosuojataapetilla

Ville Oksanen

*Eduskunnan esityslistalla on tämän kevään aikana poikkeuksellisen paljon suoraan IT-maailman arkeen vaikuttavia lainsäädäntöhankkeita. Omasta näkökulmastani kaksi on ylitse muiden: Järkeä tekijänoikeuslakiin -kansalaisaloite ja Tietoyhteiskuntakaari.*

Eduskunnan esityslistalla on tämän kevään aikana poikkeuksellisen paljon lainsäädäntöhankkeita, jotka vaikuttavat suoraan IT-maailman arkeen. Omasta näkökulmastani kaksi on ylitse muiden: Järkeä tekijänoikeuslakiin -kansalaisaloite ja tietoyhteiskuntakaari.

Näistä ensimmäinen on ollut ankaran vastalobauksen kohteena, ja aloite on koetettu leimata piraattien ja kommunistien salajuoneksi Suomen luovaa sektoria vastaan. Todellisuus ei ole ihan näin radikaali, vaan aloitteella palattaisiin suurin piirtein tilanteeseen, joka oli voimassa ennen vuotta 2006. Kyseisellä lainsäädännöllä saatiin aikaiseksi muun muassa Finreactorin tuomiot, joten mi-

hinkään piratismi-aiheeseen ei esityksen perustella olla siirtymässä.

Ehdotuksella kieltämättä rajattaisiin nykytilanteeseen verrattuna muun muassa ylläpidon vastuuta, joka koskee palvelun käyttäjien rikkomuksia. Jatkossa vastuun perusteita ei voitaisi vetää enää hihasta. Epäkohta on johtanut esimerkiksi DC++-hubien ylläpitäjille langetettuihin miljoonaluokan korvauksiin. Sen sijaan oikeudenhaltijoiden tulisi vahinkojen osalta näyttää normaalien vahingonkorvausoikeudellisten säätöjen mukaisesti, että vahinkoa on tapahtunut. Väitteet siitä, että lakialoite laillistaisi piraattipalvelut, ovat siis ihan faktisesti vääriä.

Aloitteella helpotettaisiin

myös tekijänoikeudellisten aineistojen käyttöä kouluissa. Tällä hetkellä tekijänoikeus asettaa täysin absurdeja vaatimuksia opettajille, erityisesti digitaalisessa maailmassa, ja nämä ongelmat aloite pyrkii korjaamaan. Tältä osin muutoksen ulkopuolelle on kuitenkin rajattu opetustarvittuun tehty materiaalit, millä pyritään suojaamaan kotimaista opetusmateriaalituotantoa. Rahallisesti muutoksella ei ole merkittäviä vaikutuksia (joitain miljoonia), mutta se helpottaisi todella merkittävästi digiaikaan siirtymistä, missä Suomi on jo pahasti jälkijunassa.

Aloitteen ehkä rohkein kohta on pilvipalveluiden määrittely yksityiseksi käytöksi. Jos ehdotus menisi tältä osin läpi, se tarkoittaisi, että ”piuhan pituudella ei ole merkitystä” eli kuluttajan yksityiskäytössä olevat palvelut olisivat yhtä laillisia tarjottuna verkossa kuin kotona olevasta laatikosta. Tässä kohtaa Suomi olisi edelläkävijä globaalisti, minkä voisi kuvitella lisäävän Suomen houkuttavuutta entisestään pilvipalveluiden sijaintimaana.

Tälläkään ehdotuksella ei tuhoitaisi suomalaisten sisälöntuottajien elinkeinoa. Rahoitusmalli yksityisen käytön osalta on hyvitysmaksu, joka tässä tapauksessa tarjoaisi byrokraatiasta vapaan lähestymisen palveluihin. Jos kustannustaso saataisiin painettua esimerkiksi alle 5 %:iin liikevaihdosta, kyseessä olisi erittäin hyvä tarjous globaaleille palveluntarjoajille, joita Suomeen kaivattaisiin kipeästi lisää.

Valitettavasti aloitteen tie eduskunnassa näyttää lähes epätoivoiselta. Teoston kummien ja Viestinnän keskusliiton lobbarien asemaa sivistysvaliokunnan luotettuina tiedonlähteinä on jokseenkin mahdotonta murtaa. Lex Karpelan yhteydessä tämä melkein onnistui suuressa salissa hyvin massiivisen ”masinoin-

nin” avulla. Nyt tarvittaisiin vastaava massiivinen ruohonjuuritason kampanja, jotta asian tärkeys saataisiin juntattua kansanedustajien päähän.

### Tietoyhteiskuntakaari

Eduskunnan käsittelyssä on myös tietoyhteiskuntakaari, johon on koottu kaikki viestintää koskevat, aikaisemmin erilliset lait. Paketissa ovat siis mukana muun muassa teletietojen tallennusvelvollisuus viranomaiskäyttöön, verkkopalvelun tarjoajien vastuu ja velvollisuudet sekä verkkoneutraalius.

Kokonaisuuden laajuuden vuoksi on pelättävissä, että vain muutama aihe saa eduskunnassa ansaitsemansa huomion. Lisäksi kyse on hyvin teknisestä lainsäädännöstä. Esimerkiksi verkkoneutraaliuden kohdalla tämä asettaa melkoisia vaatimuksia kansanedustajille. Heidän tulisi ymmärtää syvällisesti internetin toimintaa, jotta he voisivat ottaa perustellun kannan ehdotuksen sisältöön.

Positiivisena puolena on se, että tietoyhteiskuntakaari käsitellään lähes jokaisessa eduskunnan valiokunnassa, tulevaisuusvaliokunnasta hallintovaliokuntaan. Tilaisuuksia erilaisten näkemysten esittämiseen tulee siis olemaan hyvin runsaasti.

Eduskunnassa on toki muutakin seurattavaa. Yksi varsin mielenkiintoinen kokonaisuus on rahankeräyslain uudistushanke. Nykyinen laki on hyvin ongelmallinen erityisesti joukkorahoitettujen hankkeiden osalta. Vaikka jo tällä hetkellä tuotemyyntiin tähtäävä joukkorahoitus on sallittua, esimerkiksi avoimien kulttuurihankkeiden joukkorahoitus ei ole mahdollista. Nyt kevään aikana nähdään, korjataanko tämä ongelma. 🐛

*Kirjoittaja on tietotekniikka-erikoistunut Aalto-yliopiston tutkija ja Electronic Frontier Finland ry:n varapuheenjohtaja.*

# Visuaalisen simulaation monet muodot

*Kun simulaattoreista haetaan äärimmäistä eläytymistä, pelkkä monitori ja peliohjain eivät riitä. Lisävarusteilla ja hyvällä suunnittelulla saa ihmeitä aikaan.*

Teksti: Ville Ranki Kuvat: NASA, Wikimedia Commons, Jukka O. Kauppinen, Ville Ranki, Malmin ilmailukerho, Markus Rätty.

Visuaalisella simulaatiolla tarkoitetaan fyysisten tilojen jäljitteilyä, jossa painotetaan kuvallisia keinoja. Sen tyypillisiä sovelluksia ovat erilaisten kulkuneuvojen ja ympäristöjen simulaattorit. Tässä yhteydessä simulaattori-käsite kattaa kaiken laitteiston, jota jäljittelyyn tarvitaan, eikä siis pelkästään simulointiohjelmistoa tai -peliä. Simulaattorin tärkein varuste on visuaali, maisemakuvaa esittävä näyttöpinta.

Visuaalinen simulaatio on käsitteenä lähellä tekotodellisuutta. Tekotodellisuus kuitenkin kattaa kaikenlaiset abstraktejakin asioita sisältävät virtuaaliympäristöt. Visuaalisessa simulaatiossa simuloidaan jotakin todellista, oli se sitten surffilauta tai avaruussukkula.

Useimmat visuaalisen simulaation sovellukset ovat koulutuksessa tai tutkimuksessa käytettyjä ison rahan kulkuneuvosimulaattoreita. Tästä huolimatta visuaalinen simulaatio on nykyään mahdollista myös harrastusbudjetilla. Tietokoneiden ja näyttölaitteiden hinnat eivät ole koskaan olleet niin alhaalla kuin nyt. Kohtuullisellakin budjetilla voi rakentaa

simulaattoreita, jotka hyvin pitkälti vastaavat kaupallisia tuotteita. Harrastajapuolella hallitsevia sovelluskohteita ovat lento- ja ajosimulaatio.

Visuaalisessa simulaatiossa immersio on erittäin tärkeässä osassa. Sillä tarkoitetaan käyttäjän kokemusta siitä, että hän on fyysisesti läsnä simuloidussa maailmassa. Esimerkiksi kaupallisissa lentosimulaattoreissa immersio on lähes täydellinen, eli käyttäjä ei kovin helposti pysty erottamaan simulaatiota tosimaailmasta. Immersiota voi parantaa hyvin yksinkertaisilla keinoilla kotikonstein, joihin palataan myöhemmin tässä artikkelissa.

Tavallinen pelikokemus muuttuu elämykseksi jo kohtuullisella immersioilla. Suomessa ja maailmalla on satoja harrastelijavoimin tehtyjä simulaattoreita, ja harrastus kasvaa kovaa vauhtia. Harrastuksen voi aloittaa yksinkertaisimmillaan hakemalla kaupasta ratti-poljinyhdistelmän ja rallipenkin. Suuruudenhulluimmat vuokraavat teollisuushallin 737-simulaattorin rakentamiseen.

## Historiaa

Visuaalisella simulaatiolla on pitkälti yhteinen historia avaruus- ja lentosimulaattorien sekä varhaisten tietokonegrafiikkakokeilujen kanssa. Ensimmäisissä visuaalilla varustetuissa lentosimulaattoreissa kuva tuotettiin TV-kameralla, jota liikutettiin suuren pienoismallin yläpuolella. Samoin esimerkiksi Gemini-avaruussimulaattoreissa ulkonäkymät tuotti pienoismalleja kuvaava TV-kamera.

1960-luvulla tietokoneet tulivat mukaan kuvioihin. Alan pioneereja olivat muun muassa tietokonegrafiikkagurut Douglas Engelbart, David Evans ja Ivan Sutherland. Vuonna 1971 julkaistiin ensimmäinen kaupallinen ”yövisuaali”, jossa vektorinäytölle piirretty maisema koostui pisteistä. Pian pystyttiin piirtämään jo yksinkertaisia täytettyjä monikulmioita ja myöhemmin pintakuviotuja malleja. Kotitietokoneiden grafiikkaominaisuudet seurasivat samaa kehitystä vuosikymmenen tai pari jäljessä.

Suuri yleisö pääsi kokemaan visuaalista simulaatiota 1980-luvulla. Kotitietokoneiden grafiikkaominaisuudet olivat





Yksinkertainen kotiohjaamo.

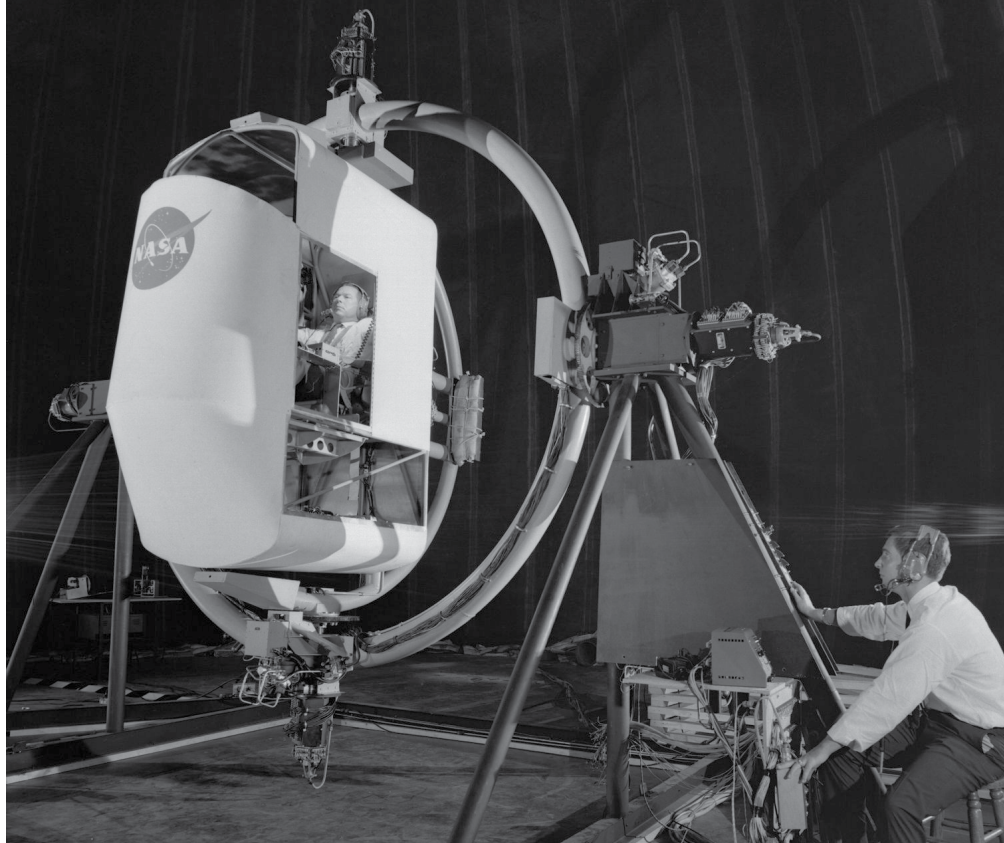


Taustaprojektiolla ja mittaristonäytöllä varustettu kotiohjaamo.

tuolloin vielä alkeellisia, mutta kolikkopelien 16-bittisillä prosessoreilla ja suurilla kuvaputkilla voitiin tuottaa uskottavaa reaaliaikaista 3D-grafiikkaa.

Atari julkaisi vuonna 1988 Hard Drivin' -autosimulaattorin, jota näki monessa pelihallissa täällä Suomessaakin. Peli oli ensimmäinen autopeli, jossa oli oikeasti realistinen fysiikkamallinnus ja täytettyä 3D-grafiikkaa. Ohjaimina siinä oli jämää force feedback -ratti, vaihteet ja kytkin. Kabinetin laidat täyttivät pelaajan näkökentän lähes kokonaan. Ruudun alareunaan piirretty mittaristo heilui kurveissa keskipakoisvoimien armoilla. Näiden yksinkertaisten temppujen avulla pelistä tuli immerstiivinen kokemus. Pari vuotta myöhemmin julkaistu jatko-osa Race Drivin' onkin kirjoittajan suosikkiautopeli vielä tänäkin päivänä.

Tämän jälkeen kolikkopeleissä trendi siirtyi muutamaa poikkeusta lukuun ottamatta helpompiin, epärealistisempiin ajomallinnuksiin ja vähemmän simulaattorimaiseen lähestymistapaan. Realismia kaipaavalle pelaajalle parhaat vaihtoehdot ovatkin 90-luvulta alkaen löytyneet PC- ja konsolipeleistä kuten F1GP-sarja, Gran Turismo, Forza Motorsport, Grand Prix Legends, X-Plane ja Flight Simulator.



NASAn simulaattori jolla harjoiteltiin telakoitumista kuun kiertoradalla vuonna 1964.



Nykyaikainen kaupallinen simulaattori sähkömekaanisella liikealustalla.

### Harrastajalle

Kotisimulaattoria himoitsevan ensimmäinen pulma on päättää käytettävä ohjelmisto, käytännössä usein peli. Jos simulaattoriin halutaan erillinen mittaristo, itse tehdyt ohjaimet tai muuta erikoisempaa rautaa, täytyy ohjelmiston tukea niitä. Erityisesti konsolipeleissä tämä saattaa tuottaa vaikeuksia.

Lentosimulaattoripuolella Flight Simulator ja X-Plane tarjoavat hyvät rajapinnat simulaation tilan lukemiseen ja

kirjoittamiseen sekä erikoisempienkin ohjaimien käyttämiseen. Nämä ovatkin suosituimmat ohjelmistot kotisimulaattoreissa. Sotilaslentosimulaattoreissa tilanne on hieman huonompi, mutta esimerkiksi IL-2-sarja, DCS: A-10 ja Falcon 4 tukevat mittariarvojen lukemista jossain määrin. Autopuolella suosittuja ovat Live for speed, rFactor, iRacing ja Forza 4.

Simulaattorin fyysisessä suunnittelussa täytyy palata immersion käsitte-

seen, sillä koko projektissahan on kyse aivojen huijaamisesta. Ihmisen tasapainoaisesti hyödyntää paljon näköaistia, joten visuaalin tulisi täyttää mahdollisimman suuri osa näkökentästä. Visuaalin ulkopuolelle jäävä alue tulisi piilottaa esimerkiksi mattamustalla levyllä tai kankaalla. Käytännössä simulaattorin ympärille kannattaa tehdä jonkinlainen koppi, joka ulottuu sivusuunnassa ja yläpuolella hieman istuimen takapuolelle ja peittää visuaalin ulkopuolelle jäävän näkökentän. Immersio kärsii, jos visuaalin vieressä näkyy sotkuinen huone.

Ihmisen stereonäöstä on visuaalin toteutuksessa pelkkää haittaa, sillä sen vuoksi käyttäjä voi huomata, että näyttöpinta on tosiasiaa lähempänä kuin maisema, jota se yrittää esittää. Visuaalin näyttöpinnan tulisikin olla mahdollisimman kaukana, mutta altaassa tilassa voi metriä tai paria pitää riittävänä minimimatkana.

Visuaalin kuvakulma on käyttäjän silmistä näyttöpinnan reunoihin vedettyjen viivojen välinen kulma. Kulman voi laskea kaavalla  $2 \cdot \arctan(\text{leveys} / \text{etäisyys} / 2)$ . Jos käytetyssä ohjelmistossa voi asettaa näkökentän laajuuden (FOV, field of view), tulisi se asettaa tähän arvoon,

jotta kappaleet näkyvät visuaalilla oikean kokoisina. Jos visuaalin pinta-ala jää pieneksi, voidaan tinkiä realismista ja käyttää suurempaa virtuaalista näkökenttää, jolloin kuva vääristyy hieman.

Simulaattoriin tulevat ohjaimet valitaan omien tarpeiden ja budjetin mukaan. Esimerkiksi autosimulaattorissa voi käyttää kaupasta ostettuja ratti-poltinjyhdistelmiä tai rakentaa ne itse oikeista auton osista. Hiiri, näppäimistö tai gamepad eivät saa olla näkyvissä.

Immersio lisääntyy mukavasti myös, jos mittarit saadaan erilleen visuaalista, mutta esimerkiksi konsolipeleissä tämä saattaa olla hankalaa. Yleinen temppu on piirtää mittarit jollain sopivalla ohjelmistolla tavalliselle monitorille ja tehdä monitorin eteen peitelevy, jossa on reiät mittarien paikoilla. Osa harrastelijoista vannoo mekaanisten mittareiden perään. Netistä löytyy hyvin ohjeita, joilla esimerkiksi oikean auton mittarikonsolin ja vaihdenäytön voi asentaa ajosimulaattoriin. Ihanteellisesti visuaalilla ei näy muuta kuin ulkonäkymä ja kaikki muu on mittaristossa omalla paikallaan.

Istuimeksi kannattaa hankkia jotain järkevempää kuin toimistotuoli. Oikea auton rallipenkki on tietysti hieno, mutta

melkein mikä vain tukeva ja mukava tuoli riittää. Kannattaa myös miettiä, miten penkin tai ohjaimet voi säätää erikokoisille käyttäjille. Vakavammat harrastajat hankkivat simulaattorin pohjaksi oikean auton tai lentokoneen rungon. Oikeita osia hankittaessa eBay ja romuttamot ovat hyviä ostospaikkoja. Äänentoistoa suunniteltaessa kannattaa noudattaa teiniautolinjaa: reilusti bassoa!

Visuaalinen simulaatio on koukuttava harrastus, jossa yhdistyvät tietotekniikka, mekaaninen rakentelu ja elektroniikka. Vaikka lento- ja ajosimulaattorit ovat nykyään lähes ainoita kotisimulaattoreita, toivon että jatkossa nähdään enemmän vaihtelua. Itse kokonaan tehdyn ohjelmiston tekeminen on hyvien peliohjelmointikirjastojen ansiosta helpompaa kuin koskaan. Sopivalla tiimillä esimerkiksi monen pelaajan panssarivaunu-, mech- tai sukellusvenesimulaattorin tekeminen tyhjästä ei siis olisi mitenkään mahdoton ajatus. Tässä artikkelissa esitettiin tarvittavat perustiedot aiheesta, ja internetistä löytyy hakemalla valtavasti itsetehtyjä simulaattoreita herättämään ajatuksia. Ei muuta kuin rakentamaan.

## Visuaalin toteutus

Visuaali on usein simulaattorin kallein osa. Resoluution ja grafiikan laadun, niin sanotun silmäkarkin, merkitys yliarvioidaan usein. Immersion kannalta resoluutiota tärkeämpää on näytön pinta-ala, eli samalla näkökentän laajuus. Iso kuva pienellä resoluutiolla on parempi kuin pieni kuva isolla. Esimerkiksi monessa kaupallisessa visuaalissa resoluutio on vain luokkaa 1024×1024 pikseliä per valkokangas.

Grafiikan laadulla ei myöskään ole suuresti merkitystä. Esimerkiksi Finnairin äskettäin eläkkeelle siirtyneessä DC-9-simulaattorissa on pisteistä koostuva yövisuaali, mutta lentokokemus on silti hyvin realistinen. Pelin grafiikka-asetukset kannattaakin ruuvata sen verran vaatimattomiksi, että ruudun päivitystaajuus säilyy riittävänä. Jatkuvan päivitystaajuuden pitäisi olla vähintään luokkaa 40–60 kuvaa sekunnissa. On tärkeää, että päivitystaajuus pysyy suurin piirtein vakiona eikä nyi havaittavasti esimerkiksi maise-  
maa ladatessa.

Visuaali tulisi sijoittaa pystysuunnassa niin, että siinä näkyvä horisonttiviiva on käyttäjän silmien korkeudella.

Seuraavaksi esittelemme suosituimpia vaihtoehtoja visuaalitekniikaksi hyvine ja huonoine puolineen.

Seuraavaksi esittelemme suosituimpia vaihtoehtoja visuaalitekniikaksi hyvine ja huonoine puolineen.

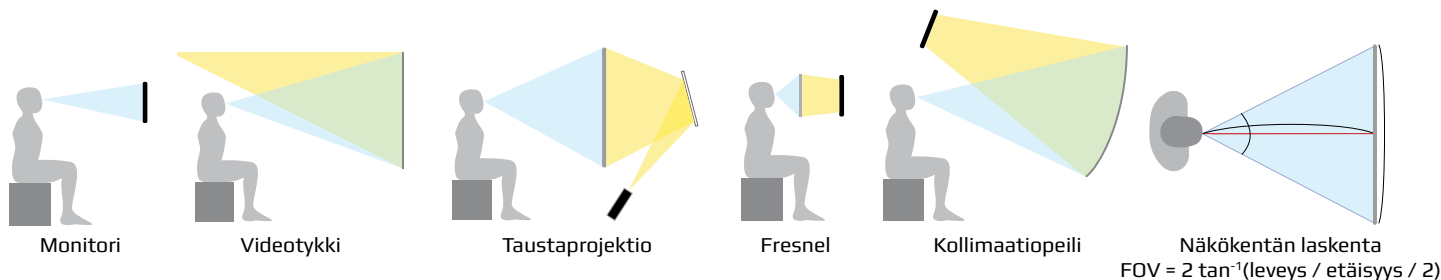
### Monitorit ja taulu-TV:t

Monitorit ovat halpoja, ja niissä on erinomainen resoluutio. PC-peleihin saa toimivan pikkuvisuaalin esimerkiksi asentamalla kolme monitoria vierekkäin. Monitorien reunat tekevät kuvaan saumat, joille ei mahda oikein mitään. Pinta-ala jää monitoreilla muihin vaihtoehtoihin verrattuna surkeaksi, ja niitä käyttäessä katseluetäisyys jää helposti

liian pieneksi. Monitorit voi myös helposti sijoittaa kaaren muotoon niin, että ne peittävät suuremman kuvakulman. Taulutelevisiot toimivat visuaalina monitoreja paremmin, mutta niiden hinnat ovat käytettyinäkin vielä melko korkealla. Resoluutio ja kuvan pinta-ala ovat kohtuulliset.

### Videotykki

Suosituin tapa toteuttaa visuaali. Kuvan pinta-ala on suuri, ja nykyään resoluutiokin on lähellä monitoria. Usealla videotykillä saa tehtyä todella suuren ja saumattoman visuaalin, mutta tämä saattaa vaatia erikoisen geometriakorjausohjelmiston käyttöä. Videotykki valkokankaineen vaatii melko ison tilan. Lisäksi ohjaamo pitää suunnitella siten, että rakenteet tai käyttäjän pää eivät varjosta valkokangasta. Jos on varaa sijoittaa hieman lisää rahaa tykkiin, kannattaa



hankkia niin sanottu short throw -tykki, joka pystyy projisoimaan täysikokoisen kuvan hyvinkin läheltä valkokangasta. Jos tykissä on hyvät geometriasäädöt, sen voi sijoittaa vapaammin.

### Taustaprojektio

Taustaprojektiossa kuva projisoidaan videotykillä valkokankaan takaa. Se on suosittu tapa saada suurikokoinen visuaali pieneen tilaan. Valkokankaan voi sijoittaa melko lähelle käyttäjää, koska varjostusongelmaa ei ole. Taustaprojektio hukkaa hieman videotykin valotehoa ja saattaa sumentaa kuvaa. Yleinen projektio-televisioissakin käytetty tempu on heijastaa kuva valkokankaan takana olevan peilin kautta, jolloin videotykin voi sijoittaa valkokankaan ylä- tai alapuolelle ja tarvittava etäisyys puolittuu.

### Fresnel-linssi

Muovista valmistettu levy, joka suurentaa monitorin kuvan ja saa sen näyttämään kaukaiselta. Fresnel-linssin voi sijoittaa lähelle katsojan kasvoja, jolloin tavallisesta näytöstä saa suuren, näennäisesti kaukana olevan kuvan. Haittapuolena fresnelin kuvaan tulee linssin rakenteesta johtuvia himmeitä sisäkkäisiä ympyröitä. Lisäksi kuva näyttää oikealta vain tarkalleen linssin edestä, joten visuaalia voi katsoa vain yksi henkilö kerrallaan. Usealla fresnel-linssillä ja monitorilla saa aikaan lähes saumattoman visuaalin, mutta sellaisen virittäminen on työlästä.

### Kollimaatiopeili

Suuri tarkasti muotoiltu kovera peili, joka heijastaa kuvan pienemmältä, yleensä katsojan yläpuolella olevalta näytöltä. Kollimaatiopeili on ikään kuin videotykillä toteutettu visuaali, mutta valkokan-

kaan tilalla on peili ja tykin tilalla näyttöpinta. Kollimaatiopeiliä käytetään isoissa lentosimulaattoreissa ja sillä saadaan aikaan laaja ja saumaton visuaali. Lisäksi kuva näyttää olevan noin kaksi kertaa niin kaukana katsojasta, kuin itse peili tosiasiaa on. Kollimaatiopeilin hinta on harrastajan ulottumattomissa, ja sellaisen tekeminen kotikonstein on hyvin vaikeaa.

### Silmikkonäytöt

Näyttötekniikoiden ”musta hevonen”. Perinteisesti harrastajahintaiset silmikkonäytöt ovat olleet surkeita, mutta Oculus Rift (katso Skrolli 2013.3) on toivottavasti muuttamassa tilannetta. Silmikkonäyttöjen ongelmana on se, että niiden läpi ei näe. Tämän vuoksi ne soveltuvat vain sellaisiin simulaattoreihin, joissa fyysisten ohjainten määrä on pieni ja käyttäjä löytää ne näkemättä.

## Muuta alan tekniikkaa

### Liikealustat

Liikealusta kallistelee tai liikuttaa koko simulaattoria yhdellä tai useammalla akselilla. Useimmiten liikealustat on toteutettu sähkömoottoreilla tai hydraulilla. Valitettavasti liikealustat ovat toistaiseksi harrastelijan ulottumattomissa korkean hintansa vuoksi. Kaupalliset mallit maksavat kymmeniä tuhansia euroja, ja liikealustan toteuttaminen itse on työläs projekti. Liikealustalle tehdyssä simulaattorissa täytyy ottaa huomioon myös sen massa ja kestävyys. Suomessa ei ole tiettävästi käytössä yhtään harrastelijoiden toteuttamaan liikealustaa.

### Tärinälaitteet

Tärinälaitteet luovat infraääniä, jotka tärisyttävät pelaajan penkkiä. Ne muistuttavat teknisesti bassokaiuttimia, ja niihin ohjataan usein peliäänä.

### Stereolasit

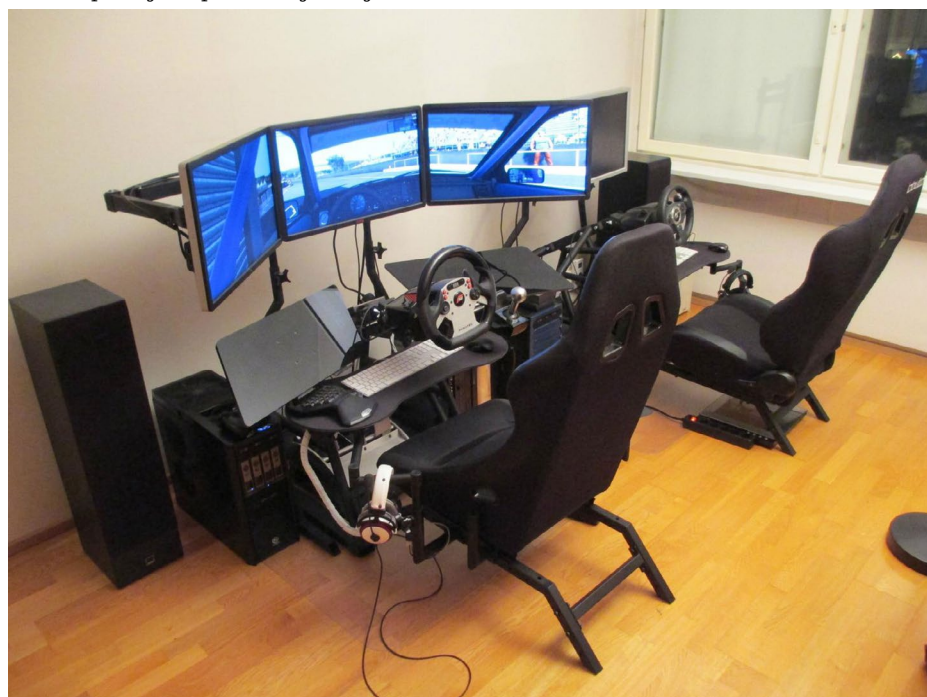
Stereolasit lienevät useimmille tuttuja 3D-elokuvateattereista. Suljinlasit peittävät vuorotellen yhden silmän samalla, kun ruudulle piirretään toiselle silmälle tarkoitettu kuva. Kun tätä toistetaan riittävän nopeasti (usein 120 Hz), saadaan toimitettua silmille eri kuva. Ihmissilmän stereonäkö toimii vain muutaman kymmenen metrin etäisyydelle, joten stereolasit ovat likimain hyödyttömät lentosimulaattoreissa. Autosimulaattoreissakin niiden hyödyn voi kyseenalaistaa. Realis-

ti- tinen stereokuva vaatii käytännössä aina tuen käytetyltä peliltä. Yleensä stereolasien mukana tulee ajuri, joka koettaa pelin piirtokäsytien perusteella arpoa stereokuvan, mutta tämä ei voi edes teoriassa toimia oikein.

### Päänseuranta

Päänseurantalaitteilla pelaaja kääntää virtuaalihahmonsansa katseen suuntaa kääntämällä fyysisesti omaa päätään. Laitteet on toteutettu useimmiten kolmella silmälasien sankoihin tai hattuun kiinnitettävällä infrapunaledillä ja tietokoneeseen kytketyllä kameralla, joka kuvaa pelaajan päätä. Ajuriohjelmisto

laskee ledien paikkojen perusteella pään paikan ja asennon. Valitettavasti kiinteiden visuaalien kanssa laitteita ei voi käyttää, sillä ulkonäkymä ei todellisuudessaakaan pyöri pelaajan pään mukana. Tämä tappaisi immersion alkuunsa. Sen sijaan silmikkonäytöissä päänseuranta toimii oikein hyvin, ja esimerkiksi Oculus Riftissä on mukana kolmen vapausasteen kiihtyvyyssanturiin perustuva päänseuranta.



Obutto-ergonomiaratkaisun päälle rakennettu autosimulaattori.

# Esimerkkejä

## CAVE

”Cave Automatic Virtual Environment” tarkoittaa immerssiivistä virtuaaliympäristöä, jossa heijastetaan videotykeillä kuva kuution muotoisen huoneen seiniin. Kirjoittaja tutustui opiskeluaikanaan Tampereen teknillisen yliopiston Virtual Reality Centerissä silloin olleeseen CAVE-laitteistoon, ja tämä esimerkki kertoo siitä.

TTY:n CAVEssa oli 3×3×3-metrinen kuution muotoinen huone, jonka katto ja kolme seinää olivat taustaprojektiopintoja. Pinnoille heijastettiin ulkopuolelta stereokuva, ja käyttäjien tuli käyttää stereolaseja. Laseissa oli paikannuslaite, jonka avulla pintojen kuva sovitettiin lasien käyttäjän näkökulmaan. Käyttäjä pystyi liikkumaan CAVEN lattialla ja esimerkiksi kurkistamaan virtuaalisen ikkunan sisään tai kurkistamaan kumartamalla jonkin esineen alapuolelle. Vaikka huone oli kuution muotoinen, sen reunat saatiin häivytettyä lähes näkymättömiksi. Kunkin seinän resoluutio oli 1024×1024 pikseliä.

Virtuaalimaailman kanssa vuorovai-  
kuttamista varten käyttäjällä oli paikan-  
nuslaitteella varustettu kaukosäätimen  
näköinen laite. Sillä pystyi laserosoit-  
timen tapaisesti osoittamaan esineitä  
virtuaalimaailmassa ja käyttämään esi-  
merkiksi virtuaalisia valikoita. CAVEN  
grafiikoita pyöritti Silicon Graphicsin  
Onyx2-tietokone, jossa pyöri IRIX-käyt-  
töjärjestelmä. Ohjelmointi tapahtui C/  
C++:lla, OpenGL:llä ja CAVEN erikois-  
ominaisuuksien toteuttamista helpotta-  
villa kirjastoilla.

CAVE on hyvä esimerkki yleiskäyttöi-  
sestä virtuaaliympäristöstä. Sen sisään  
voitiin tuoda erilaisia simulaatioita var-  
ten vaikkapa nostolavan ohjauslaitteet  
tai liikealustalla varustettu metsäkoneen  
ohjauspenkki. Yksi käyttötarkoituksista  
oli myös tieteellinen, tekninen ja arkki-  
tehtuurinen visualisointi. Vaikka CAVElle  
oli sovitettu mm. Quake, oli oma suosiki-  
ni yksinkertainen demo, jossa pystyi  
leikkimään teräsmiestä ja lentelemään  
vapaasti virtuaalikaupungissa.



## Simrac

Tamperelainen yritys Simrac valmistaa  
koulutus- ja rallisimulaattoreita. Rallisi-  
mulaattori on korvausta vastaan avoinna  
yleisölle. Sähkömekaaninen liikealus-  
ta on kiinnitetty oikean ralliauton tuki-  
rakenteisiin. Autossa on luonnollisesti  
täysin toimivat ja force feedbackilla va-  
rustetut ohjaimet. Visuaalina on suuri  
elokuvatheaterimainen valkokangas, jo-  
hon heijastetaan kuva yhdellä videoty-  
killä. Ajaminen on varsin realistista ja  
oikeasti hikistä puuhaa. Laitteisto on ke-  
hitetty ja valmistettu itse, mutta siinä on  
käytetty myös kauppoista löytyviä osia ja  
laitteita.



## MIK Cessna

Malmin ilmailukerhon lentosimulaattori  
OH-SIM on oikean lentokoneen runkoon  
rakennettu täysiverinen simulaattori kai-  
killa herkuilla. Finnairin lahjoittamien  
kolmen kollimaatiopeilin ympärille to-  
teutettu visuaali tarjoaa lentäjälle 160  
asteen näkymän ja luo aidolta tuntuvan  
syvyysvaikutelman maisemaan. Ohjaus-  
ja hallintalaitteet ovat aivan autenttisen  
oloisia – monet niistä ovatkin oikean ko-  
neen osia. Kokonaisuudessaan ohjaamo  
vastaa täysin uutta 2000-luvun Cessna  
172 -pienkonetta. Simulaattori on raken-  
nettu täysin talkootyönä.



## Ilmailumuseoiden mersut

Helsinki-Vantaan lentoaseman kupeessa  
sijaitsevassa Suomen Ilmailumuseossa  
on Virtuaalilentäjät ry:n talkoovoimin  
rakentama Messerschmitt Bf 109 -simu-  
laattori. Simulaattorin ohjaamo on jäl-  
jennös saksalaisvalmisteisen hävittäjän  
rungosta kaikkine hallintalaitteineen.  
Ohjaamon mittarit on toteutettu monito-  
rilla, jonka edessä on rei’itetty paneeli.  
Visuaalina toimii kolmella short throw  
-videotykillä huoneen etu- ja sivuseinille  
projisoitu kuva. Visuaali lienee hienoin  
Suomessa harrastevoimin rakennetuista,  
sillä se leviää reilusti ohjaamon takapu-  
olelle ja täyttää lentäessä lähes koko näkö-  
kentän. Ohjelmistona on Il-2 Sturmovikin  
muokattu versio. Myös Keski-Suomen  
Ilmailumuseossa Tikkakoskella on lähes  
samanlainen simulaattori.

## Artemis Starship Bridge Simulator

Artemiksessa joukko pelaajia miehittää  
avaruusaluksen komentosiltaa seikkail-  
len pelin Star Trek -henkissä universu-  
missa. Pelaajilla on komentosillalla jaetut  
tehtävät, ja jokainen vastaa esimerkiksi  
aluksen ohjauksesta, asejärjestelmästä,  
konehuoneesta tai viestinnästä. Jokaisel-  
la pelaajalla on oma työasema, joka luo-  
daan joko kannettavan tietokoneen tai  
tabletin ruudulle.

Yhteisellä isolla näytöllä näkyy pää-  
näkö, josta näkee yleensä ulos avaruuteen.  
Yksi pelaajista valitaan kapteeniksi, jolla  
ei ole omaa työasemaa ja joka antaa muille  
komentoja. Artemiksen hupi ja immersio  
perustuvat ihmisten väliseen kommunikoi-  
ntiin, jossa Star Trekin tuntemuksesta on  
apua. Immersiota voi vielä kasvattaa  
pukeutumalla alan asusteisiin ja sisusta-  
malla pelihuoneen scifiteemaisesti. 🚀



## TEKin jäsenyys – Työelämän toimivin käyttöliittymä

YT:t? Uusi duuni haussa? Paljonko palkkaa? Kaipaako CV:si kohennusta tai työsopimuksesi tarkistusta? Vai etsitkö urallasi uutta suuntaa? Kiinnostaako yrittäjäyys? Lähdössä ulkomaille? Haluatko verkostoitua alasi ammattilaisten kanssa?

TEK auttaa näissä ja monissa muissa urasi käännteissä. Ajamme etujasi IT-alalla, edistämme ammattikunnan osaamista ja yhteisöllisyyttä sekä rakennamme kestävää hyvinvointiyhteiskuntaa.

[www.tek.fi/liity](http://www.tek.fi/liity)

IAET-  
kassa



LAKIPALVELUT



PALKKA-  
NEUVONTA

CV-KLINIKKA  
&  
URA-  
NEUVONTA



OPISKELIJAJÄSENYYS  
ON MAKSUTON!  
VUOSIJÄSENEEN JÄSENMAKSU  
29 €/KK  
(SIS. TYÖTTÖMYYSTURVA)

**TEK**  
TEKNIIKAN AKATEEMISET

Tekniikan akateemiset TEK on tekniikan alan yliopistokoulutettujen etu- ja palvelujärjestö. Jäsenet ovat tekniikan, tietojenkäsittelyn, arkkitehtuurin tai luonnontieteiden yliopistotutkinnon suorittaneita ja vastaavissa tehtävissä työskenteleviä korkeakoulututkinnon suorittaneita sekä näiden alojen opiskelijoita. TEK tukee jäseniään työuralla, edistää ammattikunnan osaamista ja yhteisöllisyyttä sekä rakentaa kestävää hyvinvointiyhteiskuntaa. Noin 73 000 jäsenellään TEK on Akavan toiseksi suurin jäsenliitto.

# Supertietokoneet Laskentatehon linnakkeet

*Komealta kalskahtava supertietokone tuo mieleen elokuvissa vilahtelevat vuoren kokoiset numeronmurskaimet. Nykyisin supertietokoneen voi kuitenkin rakentaa vaikka itse.*

Teksti: Olli-Pekka Lehto Kuvat: Heikki Sonninen, Kristian Niinen, Erkki Makkonen, Jyrki Hokkanen, Jani Tiihonen, Atte Rusanen, Johnny Korkman, Jari Kiviaura, CSC, Wikimedia Commons

**S**upertietokoneen määritelmä on erittäin epämääräinen. Tyypillisesti supertietokoneeksi kutsutaan järjestelmää, joka tietyllä ajan hetkellä edustaa aikansa kansainvälistä huippusuorituskykyä. Jos siis koneesi pääsee viiden sadan parhaan listalle (<http://www.top500.org>), voit hyvällä omallatunnolla kutsua sitä supertietokoneeksi.

Edellä mainittu Top 500 -sivu on tunnetuin supertietokoneiden suorituskyvyn mittari. Se perustuu niin sanottuun High Performance Linpack (HPL) -testiin, jossa ratkaistaan yhtälöryhmä  $AX = B$ . Paras suorituskyky saavutetaan yleensä niin suurella yhtälöryhmällä kuin koneen muistiin järkevästi mahtuu. Kun maailman suurimpia supertietokoneita testataan, yhtälöryhmissä on yli kymmenen miljoonaa tuntematonta muuttujaa.

Viiden sadan parhaan listalla on myös poliittista merkitystä, ja toisinaan voikin olla havaittavissa suurvaltojen keskinäistä nokittelua siitä, kellä on tehokkaimman koneen valtikka.

HPL-testiä on jo pitkään kritisoitu siitä, että se ei anna kovin tarkkaa kuvaa yksittäisen koneen todellisesta suorituskyvystä oikeilla sovelluksilla. Tämä pitää erityisesti paikkansa koneissa, joita ei

ole suunniteltu perinteiseen liukulukulaskentaan. Esimerkiksi data-analyysille onkin kehitetty oma Graph 500 -listansa, joka pohjautuu epäsäännölliseen graafialgoritmiin. On olemassa myös Green 500 -lista, joka vertailee Top 500 -koneiden suorituskykyä suhteessa energiankulutukseen.

Pitkällä aikavälillä erilaiset suorituskykyä ja keskinäistä paremmuutta mittaavat listat antavat mielenkiintoista tilastotietoa supertietokoneiden suorituskyvyn kehityksestä.

## Mihin supertietokoneita tarvitaan?

Supertietokoneista puhuminen voi kuulostaa vanhanaikaiselta, mutta niiden sovellusalueet ovat nykyään laajempia kuin koskaan. On totta, että jotkut helposti pilkottavat laskentatehtävät voidaan jakaa useiden tavallisten työpöytäkoneiden suoritettavaksi. Ajatusta on hyödynnetty esimerkiksi SETI@Home-projektissa.

On kuitenkin runsaasti ongelmia, joita ei voida hajauttaa tehokkaasti. Hajautetussa laskennassa on haittana joskus se, että koneiden väliset pitkät etäisyydet aiheuttavat liian suuren viiveen tai tekevät suurten tietomäärien siirtämisestä hidasta. On siis edelleen tarvetta isoille,

raskaaseen laskentaan tarkoitetuille koneille, jotka sijaitsevat fyysisesti samassa paikassa.

Perinteisesti supertietokoneita on hyödynnetty kemiassa ja fysiikassa erilaisiin simulaatioihin. Nykyään suosiotaan ovat kasvattaneet varsinkin fuusioreaktoreiden simulointi ja erilaisten nanomateriaalien kuten grafeenin tutkimus. Cernin hiukkaskiihdyttimien tuottama valtava datamäärä hajautetaan ruudukorakenteen avulla laskentakeskuksiin eri puolille Eurooppaa. Muun muassa Helsingissä sijaitseva Fysiikan tutkimuslaitos osallistuu tähän laskentatyöhön.

Ihmisten arjessa supertietokoneet näkyvät yleensä parhaiten sääennusteissa. Ennusteiden laatimisessa tarvitaan huomattavan paljon laskentakapasiteettia, ja parantunut suorituskyky näkyy etenkin pitkän aikavälin ennusteissa. Erilaisten äärisääilmöiden kuten hurrikaanien ennustaminen on myös kehittynyt viime vuosina huomattavasti. Erittäin pitkän aikavälin ilmastoennusteet puolestaan ovat keskeinen työkalu ilmastomuutoksen tutkimisessa ja ilmastopolitiikassa.

Myös tuotteiden suunnittelussa ja testauksessa ollaan siirtymässä yhä enemmän virtuaalimalleihin. Esimerkik-



Cray C94 vuodelta 1995, CSC:llä 1995–1999.

si lentokoneiden suunnittelu on jo hyvin pitkälle tietokoneistettua. Tavoitteena on luoda lentokoneesta täydellinen virtuaalimalli, jossa kaikki rakenteet ja järjestelmät on simuloitu tarkkaan ja jota voi myös lentää.

Lääketieteessä supertietokoneilla voidaan simuloida lääkkeiden vaikutuksia ja pyrkiä poistamaan niiden haittoja. Ihmisen perimän määrittämiseen tähännyt Human Genome Sequence (HGS) -projekti toi laskennallisen bioinformatiikan ensi kertaa valokeilaan. Teknologian huimaa kehitystä kuvastaa se, että tätä kirjoitettaessa ihmisen perimän sekven-



Cray T3E vuodelta 1995, CSC:llä 1997–.

sointi on noin 2,7 miljoonaa kertaa halvempaa kuin HGS-projektissa: tuolloin projekti maksoi 2,7 miljardia dollaria, mutta nyt sen hinnaksi tulisi vaivaiset tuhat taalaa.

Myös lääkeaineiden suunnittelu on siirtynyt enemmän laskentaan ja simulaatioon. Se vähentää ihmis- ja eläinkokeiden tarvetta ja mahdollistaa esimerkiksi nopean reagoinnin uusiin epidemioihin. Supertietokoneiden kehityksen myötä voi olla mahdollista myös suunnitella lääkeaineita yksilöllisesti: potilaat voivat saada itselleen räätälöidyt lääkkeet. Eräs alan kunnianhimoisimmista hankkeista



Cray XT4 vuodelta 2006, CSC:n Louhi 2006–.

on sveitsiläinen Blue Brain, jonka tavoitteena on luoda mahdollisimman täydellinen malli ihmisaivoista.

Maailman suurimmilla valtiollisilla tiedustelupalveluilla uskotaan olevan laajat supertietokoneressit, mutta yksityiskohdista tiedetään kuitenkin hyvin vähän. Hiljattain ilmenneet tietovuodotkaan eivät ole juuri valaisseet asiaa. Kuvaavaa kuitenkin on, että Yhdysvaltain kansallisen turvallisuusviraston (NSA) suurin konesali Utahissa tarvitsee 65 megawattia sähköä, kun taas Top 500 -listan kärkipaikkaa pitävä supertietokone haukkaa "vain" vajaat 18 megawattia. Ruotsin viestitiedustelulaitos (FRA) puolestaan ylsi vuonna 2007 viidennelle sijalle Top 500 -listalla, ja se on edelleenkin Pohjoismaiden korkein sijoitus.

Suomessa supertietokoneet ovat pääasiassa akateemisen tutkimuksen käytössä. Kansalliset superkoneressit on keskitetty Tieteen tietotekniikan keskuksen (CSC), joka tarjoaa kapasiteettia ensisijaisesti yliopistojen ja korkeakoulujen tutkimusryhmille.

### Monoliiteista klustereihin

Supertietokoneiden historia on lähes yhtä pitkä kuin tietokoneiden historia. Varhaisesta historiasta on kirjoitettu runsaasti muistelmia ja artikkeleita, ja menneisiin legendoihin kuuluvat muun muassa sellaiset koneet kuin Seymour Cray, CDC, Thinking Machines ja Esko (Suomen ensimmäinen tietokone). Sen sijaan alan lähihistoriasta ja nykytilasta on kirjoitettu vähemmän, joten keskitymme tässä artikkelissa juuri siihen. Hyppäämme suoraan 90-luvulle.

Vuonna 1994 Nasan tutkijat Thomas



Cray X-MP EA/464 vuodelta 1982, CSC:llä vuodesta 1989. Poistunut käytöstä.



Cray XC30 vuodelta 2012, CSC:n Kajaanissa toimiva Sisu-klusteri 2013 lähtien.

Sterling ja Donald Becker kytkivät useita tavallisia palvelimia lähiverkolla yhdeksi loogiseksi supertietokoneeksi. Tämän, aikanaan radikaalin prototyypin nimeksi annettiin Beowulf, joka on sittemmin yleistynyt tarkoittamaan mitä tahansa vastaavanlaiseen arkkitehtuuriin pohjautuvaa superkonetta. 2000-luvun alussa Beowulf-klusterien suosio alkoi kasvaa: aluksi pienissä ja keskisuurissa järjestelmissä mutta sittemmin myös tehokkaammissa laitteistoissa, aina Top 500 -listan terävimpään kärkeen asti.

Nykyään suuri osa maailman supertietokoneista on Beowulf-arkkitehtuuriin pohjautuvia klustereita, joiden käyttöjärjestelmänä on lähes poikkeuksetta Linux ja prosessorina yleensä Intel Xeon tai AMD Opteron.

Eräs keskeinen komponentti on kytkentäverkko, jolla klusterin palvelimet eli noodit kytketään yhteen. Verkon toteutus riippuu pitkälti käyttötarkoituksesta. Jos laskenta on helposti hajauttavissa itsenäisiin osiin, voi yhden tai kymmenen gigabitin Ethernet olla aivan riittävä. Jos kyseessä on vaativampi hajautus, joka vaatii paljon tiedonsiirtoa ja useita tietokoneita, on ratkaisuna useimmiten Infiniband-verkko. Infinibandin tuoreimman version tiedonsiirtonopeus 56 Gbit/s, ja sen viiveet ovat huomattavasti Ethernetiä pienemmät.

Muutammat yritykset (Cray, NEC ja IBM) kehittävät edelleen superlaskentaan tarkoitettuja tietokoneita, mutta niissäkin on yhä enemmän siirrytty massatuotettuihin komponentteihin, Linuxiin sekä avoimen lähdekoodin työkaluihin. Myös arkkitehtuuri muistuttaa hyvin paljon klustereita. Esimerkiksi CSC:nkin

käyttämä Cray XC30 on pohjimmiltaan x86 Linux -klusteri. Koneen tehoa lisää Crayn kehittämä Aries-kytkentäverkko, joka on suorituskyvyltään vielä Infinibandin tehokkaampi.

Muutos kuitenkin jatkuu. Viime vuonna Cray myi Aries-verkkoteknologiansa Intelille, eikä superkonekäyttöön suunnitellulle IBM:n Blue Gene -sarjalle ole tiedossa ainakaan suoraa jatkoa.

Standardeihin perustuvien klustereiden sekä Linuxin ja avoimen lähdekoodin ohjelmien yleistymisen on ollut erinomaista ainakin supertietokonekustusten näkökulmasta. Se vähentää riippuvuutta yksittäisestä laitevalmistajasta ja mahdollistaa koneiden räätälöinnin omiin tarpeisiin. Kolikon kääntöpuolena on se, että vielä 90-luvulla vallinnut eksotisten laitearkkitehtuurien monimuotoisuus on pitkälti kadonnut.

Joitakin oman tiensä kulkijoita vielä on. Esimerkiksi SGI (ent. Silicon Graphics) kehittää edelleen Ultraviolet-koneita, joissa on erittäin suuri jaettu muistiavaruus. NEC puolestaan jatkaa perinteikstä SX-sarjaa, joka pohjautuu niin sanottuihin vektoriprosessoreihin. Craylla puolestaan on oma data-analyysiin suunniteltu ja erikoisprosesseilla varustettu Urika-järjestelmänsä.

### Kiihdyttimet tulevat

Viime vuosien aikana myös erilaiset kiihdytinprosessorit ovat tarjonneet kaivatun piristysruiskeen superkoneille. Tässä kiihdytinprosesseilla tarkoitetaan erilaisia laskentaan suunniteltuja apuprosessoreja. Ajatus ei ole uusi: jo vuonna 1980 Intel toi markkinoille 8087-prosessorin, joka toimii 8088-prosessorin apu-

na ja keskittyy liukulukujen laskemiseen. Sittemmin ominaisuus sulautui itse prosessoriin ja kiihdyttimet siirtyivät marginaaliin. Yrittäjiä on ollut paljon, mutta menestys on vaihdellut.

Esimerkiksi IBM kokeili onneaan tekemällä supertietokoneversion Cell-prosessorista, joka oli kehitetty Playstation 3:ä varten. Vuonna 2008 uusi prosessori otettiin näyttävästi käyttöön silloisessa tehokkaimmassa supertietokoneessa, Los Alamosin kansallislaboratorion Roadrunnerissa. Se oli ensimmäinen kone, joka ylsi suorituskyvyssään petaflop-luokkaan. Cell oli kuitenkin erittäin hankalasti ohjelmitava, eikä se juurikaan yleistynyt.

Todellinen läpimurto tuli kuitenkin yllättävästä suunnasta: pelikäyttöön suunnatuista näytönohjaimista. Grafiikka-prosessorien (Graphic Processing Unit, GPU) kehitys oli vuoteen 2005 mennessä huimaa, koska näytönohjainvalmistajien välillä vallitsi kova kilpailu ja pelikansan lompakot olivat riittävän paksuja. Grafiikkaan tarkoitettujen suorittimien kehittyvät jopa nopeammin kuin tavalliset keskussuorittimet (Central Processing Unit, CPU).

Supertietokoneiden kehittäjät alkoivatkin selvittää, miten GPU:t voitaisiin valjastaa laskentaan. Syntyi GPGPU-konsepti eli "General Purpose Processing on GPU". Alustavat tulokset olivat lupaavia, vaikka ohjelmointi olikin erittäin kömpelöä. Myös GPU-laskennan epätarkkuus aiheutti omat ongelmansa: peleissä pikselin sijaintia ei tarvitse määrittää miljar-disosien tarkkuudella.

Eräät GPU-valmistajat, etupäässä Nvidia ja ATI, huomasivat markkinamah-





NEC SX-9 vuodelta 2008, Japanin hallituksen Earth Simulator -projektissa. Käyttö avattiin 2012.

dollisuudet ja alkoivat kilvan kehittää omia, erityisesti laskentaan soveltuvia versioita grafiikkasiruistaan. Niitä olivat Nvidia Tesla ja ATI Firestream. Nykyään myös useimpia tavallisia kuluttajakäyttöön suunnattuja GPU-piirejä voi käyttää laskentaan. Ne ovat kuitenkin ominaisuuksiltaan, laskentateholtaan ja muisin määrältään heikompia kuin nimenomaisesti laskentaan tarkoitettujen suorittimien. Perusarkkitehtuuri on silti sama.

GPU:t ovat kasvattaneet osuuttaan Top 500 -listalla ja vakiintuneet nykyään osaksi supertietokoneiden valtavirtaa. Nvidia on selvä markkinajohtaja, mikä johtuu pitkälti siitä, että se panostaa ohjelmointiympäristöön ja kehittäjien tukemiseen. AMD:n ja ATI:n suorittimet ovat kilpailukykyisiä, mutta sekavat kehitystyökulut sekä suppea sovellus- ja kirjastojarjonta ovat pitäneet ne marginaalissa. Molempien yritysten pääasialliset markkinat ovat kuitenkin tavallisten kuluttajien raudassa.

### Grafiikan ehdoilla

Nykyään grafiikkaprosessoreissa on noin seitsemän miljardia transistoria, kun taas tehokkaimmissa palvelinprosesoreissa vain kolme miljardia. Ero on siis jo yli kaksinkertainen. Grafiikkasirujen teoreettinen suorituskyky on 5-10 kertaa CPU:ita parempi, ja energiankulutuksessa ne ovat 3-5 kertaa taloudellisempia. Tuoreimman Green 500 -listan kärkipään koneet hyödyntävätkin poikkeuksetta GPU-pohjaisia kiihdyttäimiä.

Suorituskyvyn ero selittyy muun mu-

assa sillä, että GPU:ssa ei ole läheskään niin paljon ohjauslogiikkaa kuin CPU:ssa ja lähes kaikki pinta-ala on omistettu laskentayksiköille. Tämä tosin aiheuttaa sen, että prosessori ei voi suorituksen aikana yhtä älykkäästi peittää huonosti optimoidun koodin ongelmia. Niinpä ohjelmoijalla ja kääntäjällä on suurempi vastuu koodin optimoinnista.

Siinä missä normaalissa CPU:ssa on yhtenäinen välimuisti, GPU:ssa se on hajautettu pieniin paikallismuisteihin. Tämä yksinkertaistaa prosessoria ja helpottaa arkkitehtuurin laajentamista yhä suurempiin ydinmääriin. Tässäkin vastuu välimuistin tehokkaasta käytöstä siirtyy kääntäjälle ja ohjelmoijalle.

GPU:n keskusmuisti on hyvin nopeaa GDDR-muistia, joka tarjoaa nykyään noin neljä kertaa suuremman kaistanleveyden kuin CPU:n DDR-muistit (vrt. 200 Gt/s ja 50 Gt/s). GDDR-muisti on kuitenkin melko kallista ja sitä on vähemmän. Data pitää myös siirtää ensin kortille käyttämällä hitaampaa (8 Gt/s) PCI Express -väylää. AMD on kyllä suunnitellut Playstation 4:ää varten mielenkiintoisen prosessorin, jossa sekä CPU että GPU käyttävät samaa GDDR-muistia. Tuotteesta ei toistaiseksi ole tiedossa "avointa" versiota.

Jotta ohjelma toimisi hyvin grafiikka-suorittimissa, koodin pitäisi olla sellaista, että se käsittelee suuria yhtenäisiä data-määriä ja sisältää vain vähän if-ehtoja. Esimerkiksi Top 500 -listan matriisiratkaisu soveltuu hyvin GPU:ille, mutta esimerkiksi epäsäännöllisen verkoston käsittely (esim. kielitieteissä) on vaike-

ampaa.

GPU-laskenta on siis monissa tapauksissa erittäin tehokasta, mutta se ei sovellu kaikkiin käyttötarkoituksiin. Lisäksi GPU-ohjelmointi on edelleen melko vaativaa, vaikkakin se on helpottunut huomattavasti viime vuosina.

### Parhaat puolet molemmista

Intel on lähtenyt mukaan kiihdytinkilpailuun Xeon Phi -prosessoreilla, jotka pohjautuvat Many Integrated Core (MIC) -arkkitehtuuriin. MICin juuret ovat vuonna 2008 esitellyssä Larrabeenimisessä GPU-arkkitehtuurissa, jossa suuri osa GPU-logiikasta oli toteutettu x86-ytimillä. Arkkitehtuurista ei koskaan tullut kaupallista tuotetta GPU-käyttöön, mutta superkoneyhteisö heräsi: tässä olisi mahdollisesti tuote, joka tarjoaa GPU-tasoisien suorituskyvyn yhdistettynä x86-yhteensopivuuteen.

Paria vuotta myöhemmin Intel esiteltikin arkkitehtuurin uudestaan MIC-nimellä ja hiottuna erityisesti laskentakäyttöön. Ensimmäiset Phit tulivat markkinoille vuonna 2013, ja ne ovat suorituskyvyltään samaa luokkaa GPU:iden kanssa.

Fyysisesti Phi on PCI Express -kortti, joka asennetaan palvelimeen. Se poikkeaa kuitenkin grafiikkakäyttöön tarkoitetuista korkeista monin tavoin. Itse prosessorissa on 50-60 kappaletta x86-ydintä, jotka pohjautuvat alkuperäiseen Pentium-arkkitehtuuriin (P54C). Ytimeen on kuitenkin tehty monia uudistuksia, merkittävimpänä 512-bittinen liukulu-



IBM Blue Gene vuodelta 2004.

kuyksikkö. Tämän johdosta ydin ei ole täysin binääriyhteensopiva tavallisten x86-prosessorien kanssa, mutta tuleva AVX-512-käskykanta lupaa yhteensopivuutta.

Muisti näkyy yhtenäisesti kaikille Xeon Phi -prosessorin ytimille, ja välimuistien hallinta on automatisoitu. Kortissa itsessään on riisuttu Linux-käyttöjärjestelmä, johon on mahdollista kirjautua sisään ja jossa voi ajaa sovelluksia suoraan. Seuraavan sukupolven Xeon Phi (koodinimi Knight's Landing) tulee oheiskortin lisäksi tarjolle myös itsenäisenä prosessorina.

### Muita arkkitehtuureja

Phin ja GPU:iden lisäksi markkinoilla on muitakin mielenkiintoisia arkkitehtuureja. Kickstarter-projektissa syntynyt Adaptevan Parallela-suoritin on samantyyppinen manycore-arkkitehtuuri kuin edellä mainittu Xeon Phi. Siinä on runsaasti pieniä, vähän energiaa tarvitsevia ytimiä. 16-ytiminen perusmalli maksaa ainoastaan 99 dollaria. Kovin mahtavaksi supertietokoneeksi siitä ei ole, mutta se voi olla hyvä alusta rinnakkaisohjelmoinnin opettelun.

Ohjelmoitavat FPGA-piirit (Field Programmable Gate Array) soveltuvat hyvin tehtäviin, joissa ei ole kyse tavallisesta liukulukulaskennasta. Tekstin tunnistaminen ja DNA-sekvenssianalyysi ovat hyviä esimerkkejä. Haittapuolena on FPGA-piirien ohjelmoinnin hankaluus.

## Miten supertietokoneita ohjelmoidaan?

### Kielet

Supertietokoneiden ohjelmoinnissa perinteiset ohjelmointikielet kuten C ja

Fortran ovat olleet suosituimpia jo vuosikymmenten ajan. Vaikka erityisesti Fortran voi kuulostaa antiikkiselta, sillä on aktiivinen käyttäjäkuntansa tieteellisessä laskennassa. Laskennan näkökulmasta sillä on runsaasti hyviä ominaisuuksia, esimerkiksi tuki kompleksiluvuille ja moniulotteisille taulukoille. Suorituskykyä heikentäviä ominaisuuksia on pyritty välttämään. Ei ole mitenkään poikkeuksellista, että saman ohjelman Fortran-versio pyyhki C-versiolla lattian.

Kielen uusin versio, Fortran 2008, on ominaisuuksiltaan täysin moderni ohjelmointikieli, vaikka monet vanhan liiton superkonekoodaajat edelleen vannovatkin lähes 40 vuotta vanhan Fortran 77 -standardin nimeen. Siinä kun ei ole mitään ylimääräistä eikä suorituskyvylle haitallista.

Nykyään GNU/Linux-käyttöjärjestelmissä mukana tulevat GNU-kääntäjät ovat erittäin kilpailukykyisiä kaupallisten kääntäjien kanssa (mm. Intel, PGI). Erityisesti GNU Fortran on kehittynyt huomattavasti viimeisen viiden vuoden aikana.

Kääntäjät eivät kuitenkaan aina osaa optimoida koodia riittävän hyvin, ja siksi suorituskyvyn kannalta kriittisissä rutiineissa käytetään myös assembleria. Esimerkiksi keskeisimmässä lineaarigebran rutiineja sisältävissä kirjastoissa (mm. Openblas) on runsaasti eri prosessoreille optimoituja osia. Assemblerin käyttö tehostaa koodia mutta hankaloittaa sen siirtämistä uusille alustoille.

Perinteisten kielten rinnalle on noussemassa joukko korkeamman tason kieliä, joista suosituimmat ovat avoimen lähdekoodin Python ja kaupallinen Matlab. Korkeamman abstraktiotason tason kielet mielletään usein tehottomiksi, mutta taitavasti ohjelmoimalla voi niil-

län toteuttaa varsin tehokkaita ohjelmia. Pythonille on kehitetty paketit Scipy (Scientific Python) ja Numpy (Numerical Python), jotka lisäävät Pythoniin tehokkaita rutiineja taulukoiden käsittelyyn.

### Rinnakkaistaminen

Rinnakkaistamisella tarkoitetaan sitä, että laskentatehtävä pilkotaan pienempiin osiin ja osatehtävät suoritetaan samanaikaisesti. Useat prosessorit saadaan näin tehokkaammin käyttöön. Rinnakkaistaminen on ollut superkoneiden ohjelmoinnissa suurimpia haasteita 60-luvulta saakka. Viime vuosina tekniikka ja ohjelmointi on tullut tutummaksi, koska tavallisissa tietokoneissakin on usein moniydinprosessori.

Superkonekäyttöön tarkoitettujen ohjelmien rinnakkaistamiseen käytetään tyypillisesti Message Passing Interface (MPI) -ohjelmointimallia. Se perustuu melko yksinkertaisiin kirjastokutsuihin, joilla välitetään viestejä kahden tai useamman prosessin välillä. Ohjelmointimalli on erittäin tehokas mutta myös haastava, etenkin jos halutaan tehdä tehokasta ja helposti toiselle arkkitehtuurille sovittavaa koodia.

Standardista on lukuisia erilaisia toteutuksia MPI-kirjastojen muodossa. Esimerkiksi Craylla on oma, Aries-kytkentäverkkoa varten optimoitu kirjastonsa. OpenMPI puolestaan on suosittu yhteisövetoisesti kehitetty avoimen lähdekoodin MPI-kirjasto, joka toimii useiden erilaisten kytkentäverkkojen kanssa.

Jos riittää, että ohjelma suoritetaan rinnakkaisesti vain yhden noodin sisällä, on myös OpenMP varteenotettava vaihtoehto. Nimestään huolimatta sillä ei ole mitään tekemistä OpenMPI:n kanssa. Joissakin tapauksissa, esimerkiksi hyvin suurissa laskentatehtävissä, voi olla kannattavaa yhdistää molemmat niin, että OpenMP:tä käytetään noodien sisällä ja MPI:tä niiden välillä.

### Supertietokoneiden tulevaisuus

On varmaa, että supertietokoneiden suorituskyky jatkaa tulevaisuudessakin kasvamistaan ja että niiden käyttö lisääntyy. Vasta pieni osa mahdollisuuksista on hyödynnetty: on runsaasti erilaisia tutkimusalueita ja toimialoja, joilla superkoneita voitaisiin hyödyntää uusilla tavoilla.

Superkoneiden ongelmana ovat virrankulutus ja jäähdytystarve, jotka kasvavat nopeammin kuin suorituskyky. Jos nykyinen kehityssuunta jatkuu, tehokkaimmat koneet tulevat vaatimaan hurjia määriä sähköä.

Kansainvälinen superkoneyhteisö on

asettanut tavoitteekseen eksaflopin laskentatehon saavuttamisen noin vuonna 2020. Jos siinä onnistutaan, ollaan melko hyvin Mooren lakiin perustuvalla kehityskäyrällä. Nykyisellä kehityksellä tällaisen koneen sähköteho tulee olemaan noin 200 megawattia. Tarkoitus olisi kuitenkin saada sähkönkulutus supistettua 20 megawatin paikkeille, joka on vain hieman enemmän kuin mitä tämän päivän tehokkaimmat koneet kuluttavat. Ongelman ratkaiseminen vaatii radikaaleja uudistuksia niin teknologiaan, ohjelmointiin kuin infrastruktuurinkin.

### Mobiiliprosessorit

Eräs mahdollinen osaratkaisu voi löytyä mobiiliprosessoreiden maailmasta, jossa energiataloudellisuus on keskeisin tavoite. Toistaiseksi prosessorit ovat kuitenkin olleet arkkitehtuuriltaan sellaisia, että ne soveltuvat huonosti raskaaseen laskentaan ja suurten datamäärien käsittelyyn.

Arkkitehtuurit kuitenkin kehittyvät, ja tilanne on korjaantumassa. Euroopan unionin Mont Blanc -tutkimusprojektin tavoitteena on toteuttaa ensimmäinen todella kilpailukykyinen mobiiliprosessoriin perustuva laskentaklusteri. Myös Nvidialla on Project Denver -niminen hanke, jonka tavoitteena on yhdistää 64-bittinen ARM-ydin ja Nvidian GPU.

### Suoritinpinot

Mikropiirejä voidaan pinota päällekkäin kolmiulotteisiksi prosessoreiksi. Esimerkiksi muistipiirejä voi asettaa suoraan prosessorin päälle, jolloin muistikaista moninkertaistuu, muistin viiveet lyhenevät ja energiankulutus vähenee.

Tällainen teknologia on parhaillaan siirtymässä testilaboratoriosta todellisuuteen. Tänä vuonna Micronilta on tulossa ensimmäiset kolmiulotteisesti rakennetut Hybrid Memory Cube (HMC) -muistimoduulit. Heillä on myös Automata-niminen prototyyppi, johon on yhdistetty prosessointikerroksia. Tulevaisuuden kiihdyttimiin (Nvidia Volta, Intel Knight's Landing) kaavaillaan prosessorin päälle pinottavaa nopeaa muistia.

### Optiset komponentit

Suurin osa superkoneen energiasta kuluu datan siirtoon komponenttien välillä ja niiden sisällä. Siirtymällä sähköisistä signaaleista optisiin - ensin piirilevyllä ja myöhemmin jopa mikropiirien sisällä - saadaan merkittäviä energiasäästöjä. Muun muassa IBM ja HP kehittävät molemmat aktiivisesti prototyyppijä tällä saralla.

### Kvanttiprosessorit

Kvanttien superpositioon perustuva kvanttilaskenta mahdollistaa eräitä tehokkaita operaatioita. Monimutkaisesta laskutoimituksesta voidaan käydä samanaikaisesti kaikki ratkaisuvaihtoehdot läpi ja oikea vastaus voidaan poimia ratkaisujen joukosta.

Menetelmä soveltuu erittäin hyvin esimerkiksi epäsymmetrisen salauksen murtamiseen, joka on perinteisten laskentamenetelmien näkökulmasta lähes mahdoton tehtävä. Toisaalta on myös ongelmia, joita on hyvin vaikeaa tai mahdotonta muuttaa kvanteille sopivaan muotoon. Kvanttiprosessorit eivät sovellu ihan kaikkeen.

Yritys nimeltä D-Wave on kehittänyt kvanttikoneita jo useita vuosia mutta saanut melko skeptisen vastaanoton. Tutkijat kiistelevät edelleen siitä, ovatko D-Waven laitteet aitoja kvanttitietokoneita. Yhtiö on kuitenkin toimittanut laitteita muun muassa Lockheedille ja Googlen Quantum Artificial Research Labille.

### Ohjelmointimallit

Tulevaisuuden eksaflop-teholuokan koneissa tulee todennäköisesti olemaan satoja miljoonia ytimiä. Ohjelmien rinnakkaistaminen tälle tasolle vaatii uudellaisia ohjelmointimalleja. Myös energiankulutus pitää ottaa entistä paremmin huomioon ohjelmia suunniteltaessa. Näin suurissa laitteissa myös komponentteja rikkoutuu kymmenittäin joka päivä, joten ohjelmien täytyy osata käsitellä laitteistovikoja. Ne eivät saa kaatua heti ensimmäiseen pieneen laitevikaan, ja suuremmatkaan ongelmat eivät saa aiheuttaa kovin pahoja viivästyksiä.

### Infrastruktuuri

Konesalit ovat siirtymässä sinne, missä niitä on kokonaistaloudellisesti järkevintä ylläpitää. Googlen Hamina-projekti on eräs tunnetuimmista esimerkeistä, mutta myös CSC on siirtänyt raskaan laskennan Espoosta Kajaaniin, entiseen paperivaraan. Energia on halpaa, konetta voi jäädyttää avaamalla ikkunan (ainakin melkein) ja tilaa riittää.

Tulevaisuudessa suuntaus varmasti jatkuu ja kansallisista supertietokonekeskuksista siirrytään ainakin laitteistojen osalta yhä massiivisempiin kansainvälisiin keskuksiin, joiden sijainti on erittäin tarkasti valittu. 🏠

## Rakenna oma klusteri

Oman pienen laskentaklusterin rakentaminen on helpompaa kuin koskaan. Kaikki keskeiset toiminnot on toteutettavissa avoimen lähdekoodin ohjelmilla - samoilla ohjelmilla, joita maailman suurimmat tietokonekeskuksetkin käyttävät. Alla on lista tärkeimmistä komponenteista.

### Rauta

- Vähintään yksi edusta- tai hallintakone. Isoissa järjestelmissä voi olla kymmeniä hallintakoneita erilaisissa rooleissa, muun muassa sisään kirjautumiselle, etähallinnalle, eräajojärjestelmälle jne.
- Useita koneita laskentakoneiksi.
- Jonkinlainen kytkentäverkko, jolla noodit yhdistetään.

### Ohjelmisto

- **Käyttöjärjestelmäksi** GNU/Linux. On makuasia, mitä jakelukokonaisuutta käyttää, mutta tavallisimmat ovat Red Hat, CentOS, SuSE, Ubuntu ja Debian. Eräät tärkeät tieteellisen laskennan kaupalliset sovellukset tukevat ainoastaan Red Hat -tyyppistä jakelua, minkä vuoksi suuret laskentakeskukset käyttävät tyypillisesti Red Hat Enterprise Linuxia tai sen ilmaisversiota CentOS:ää.
- **Klusterihallinta** on järjestelmä, joka hallinnoi klusteria, esimerkiksi noodien asennusta. Pienessä järjestelmässä se on vapaavalintainen, suuremmissa pakollinen. Vaihtoehtoja: Warewulf, xCAT ja Rocks Linux.
- **Eräajojärjestelmä** delegoi vapaita noodeja käyttäjille ja seuraa muun muassa töiden etenemistä. Jos järjestelmässä on monta käyttäjää tai useita töitä suoritetaan samanaikaisesti, kannattaa käyttää eräajojärjestelmää. Slurm (Simple Linux Resource Manager) on monipuolinen ja melko helposti asennettava. Muita ovat esimerkiksi SGE, Torque, PBS ja LSF (kaupallinen).
- **Kääntäjät:** GNU C/C++, Fortran ja Python ovat hyvä perussetti. Muita ovat esimerkiksi LLVM, PGI (kaupallinen) ja Intel Composer XE (kaupallinen).
- **MPI-kirjasto:** OpenMPI:ssä on ilmaisista vaihtoehdoista monipuolisin tuki erilaisille kytkentäverkoille. Muita ovat esimerkiksi Intel MPI (kaupallinen), MPICH2 ja MVAPICH2 (vain Infiniband).

Alan suurimmassa konferenssissa, Yhdysvalloissa vuosittain järjestettävässä Supercomputingissa (<http://supercomputing.org/>) on Student Cluster Competition, jossa opiskelijajoukkueet kilpailevat siitä, kuka pystyy rakentamaan tehokkaimman klusterin. Toistaiseksi Suomesta ei ole vielä ollut joukkuetta. Olisikohan aika korjata tilanne?

# Analogisesta digitaaliseen

Martti Häikiö ja Essi Ylitalo: *Bit Bang. Yrjö Neuvo ja digitaalinen vallankumous. SKS 2013.*

Teksti: Jussi Määttä

Lettiin 1950-luvun alkupuolta, kun koulupoika Yrjö Neuvo rakensi ystävänsä kanssa ensimmäisen radiovastaanottimensa, vanerilevylle askarrellun yksinkertaisen kidekoneen. Vuosikymmenen puolivälissä pojat etenivät transistoriradioihin, heti kun osia sai Vaasaan tilattua. He rakensivat luultavasti kaupungin ensimmäiset taskuradiot, jotka sitten asennettiin polkupyöriin.

Vuonna 2006 samainen Yrjö Neuvo jäi eläkkeelle Nokialta oltuaan erilaisissa teknologiajohtajan tehtävissä yli kymmenen vuotta. Ennen Nokian-vuosiaan Neuvo teki pitkän ja menestyksekkään yliopistouran.



## Elämäkerta sekä ikkuna menneisyyteen

Neuvo on ollut koko ikänsä innostunut uusimmasta teknologiasta. Martti Häikiön ja Essi Ylitalon kirjoittama, viime vuonna ilmestynyt opus *Bit Bang* seuraa Neuvon elämää rinnakkain digitaalisen tekniikan kehityksen kanssa. Teoksen pääpaino on ensin yliopistourassa digitaalisen signaalinkäsittelyn alalla 70-90-luvuilla, ja vuodesta 1993 alkaen kirja toimii ikkunana Nokian nousuun.

Häikiö ja Ylitalo ovat ripotelleet tekstiin kiinnostavia teknisiä anekdootteja Neuvon uran varrelta. Esimerkiksi Tek-

nillisessä korkeakoulussa vuonna 1968 Neuvo teki diplomityön, johon sisältyi tarkan digitaalisen kellon rakentaminen. Sen toteutuksessa yhtenä ongelmana oli saada mekaaniset numerorullat pyörähtämään tarpeeksi nopeasti luvusta 59 takaisin nollaan lukujen 60-99 kautta.

Lisensiaatintyössään Neuvo toteutti lentokoneiden silloisten suunnistuslaitteiden tarkistusmittarin, jonka osana käytettiin ROM-piireihin tallennettua sinifunktion arvotaulukkoa. Samainen esilaskentakikka on perustemppu esimerkiksi peli- ja demo-ohjelmoinnissa, mutta sen rautapohjaiseen toteutukseen en ole ennen törmännyt!

Väitöskirjansa Neuvo teki Yhdysvalloissa Cornellin yliopistossa 70-luvun alussa nuorena perheenisänä. Hiukan harmillisesti tästä elämänvaiheesta on kirjassa kerrottu vain lyhyesti, vaikka Amerikan-ajoista olisi voinut kuvitella löytyneen enemmänkin muisteltavaa.

## Mikrohiiriä ja mediaanisuodattimia Tampereella

Väitelyään tohtoriksi vuonna 1974 Neuvo palasi Otaniemeen, ja jo kahden vuoden kuluttua hänestä leivottiin elektronikan professori Tampereen teknilliseen yliopistoon. Neuvo käsitti tuloaan tekevien mikroprosessorien merkityksen ja järjesti aiheesta täydennyskoulutusta insinööreille ympäri Suomea. Kurssilla opeteltiin muun muassa Intel 8080- ja Motorola 6800 -suorittimien konekieliohjelmointia. Kansainvälistä huomiota keräsivät myös Neuvon opiskelijoiden tekemät ”mikrohiiret” eli pienehköt robotit, jotka kilpailivat labyrinteissa ja tekivät muutakin hauskaa.

Tampereella Neuvon digitaalisen signaalinkäsittelyn tutkimusryhmä tuli tunnetuksi etenkin työstään mediaanisuodattimien ominaisuuksien ja niiden variaatioiden tutkimuksessa. Yksinkertaisimmillaan mediaanisuodatinta voi ajatella vaihtoehtona lineaarisille suodat-

timille, erityisesti keskiarvosuodattimelle.

Ajatellaan vaikka tilannetta, jossa saadaan näytteitä kohinaisesta signaalista. Silloin laatua voidaan yrittää parantaa ottamalla aina keskiarvo esimerkiksi viidestä viimeisimmästä näytteestä. Vastaava mediaanisuodatin ottaisi sen sijaan viidestä arvosta suuruusjärjestyksessä keskimmäisen. Neuvon ryhmän tieteelliset julkaisut esimerkiksi vektoriarvoisesta ja painotetusta mediaanisuodattimesta ovat keränneet huomattavan määrän viittauksia muilta tutkijoilta, mitä pidetään yhtenä korkeatasoisen tutkimuksen merkinä.

## Akatemia vai Nokia?

Uransa suurimman hyppäyksen Neuvo teki vuonna 1993. Hän tarttui Jorma Ollilan tarjoukseen ja siirtyi Nokian teknologiajohtajaksi. Päätös yliopistomaailman jättämisestä oli vaikea, varsinkin kun Neuvolle oli juuri järjestetty pysyvä tutkijaprofessorin virka Suomen Akatemiasa. Parempaa yliopistojopia ei maassamme tutkijoille tunneta ja nimitykset ovat yleensä määräaikaisia. Innokas ja mutkaton Neuvo sopeutuikin suuryrityksen kuvioihin helposti ja tuli firman sisällä tunnetuksi uusien, maailmalta hankittujen



Mikrohiiriä ihmeteltiin televisiossakin. Yrjö Neuvon opiskelijoiden rakentamat mikrohiiret esiintyivät Ylen Kasmasiini-ohjelmassa vuonna 1983. Jakson voi katsoa osoitteessa [http://yle.fi/elavaarkisto/artikkelit/mikrohiiret\\_kasmasiinissa\\_43217.html](http://yle.fi/elavaarkisto/artikkelit/mikrohiiret_kasmasiinissa_43217.html).

vempaimien esittelijänä, joka ei epäröinyt kaivaa kassistaan ruuvimeisseliä laitteiden tarkempaa tutkimista varten.

Nokialla Neuvon tehtäviin kuului mieltä, miltä tulevaisuuden maailma näyttää. Häikiön ja Ylitalon esiin kaivamien haastattelujen ja esitelmien perusteella vaikuttaakin, että vuosina 1996–1997 Neuvo ennakoiki kosketusnäytöllisen kamerakännyköiden suosion ja erilaisten sijaintia hyödyntävien mobiilipalveluiden tulon.

Kirjan lopussa käsitellään vielä Yrjö Neuvon tekemisiä Nokialta lähdön jälkeen. Lukijalle selviävät sekä kirjan nimen alkuperä että Neuvon tuoreimmat tulevaisuuden ennustukset. Paljastuupa myös, että 2000-luvulla Neuvo on innostunut traktoreista ja erityisesti

suomalaisista Avant-pienkuor- maajista, joiden kehittä- seenkin hän on sittemmin osal- listunut.

### Kirjan anti

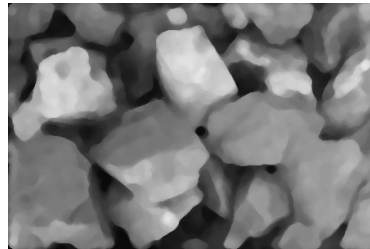
Bit Bang on sujuvasti kirjoit- tettu ja viihdyttävä elämäker- ta, joka säilytti mielenkiintoni loppuun saakka. Joitakin köm- mähdyksiä on teknisten aiheiden käsittelyssä ja suomenkie- listen termien käytössä, mutta eivät ne lukukokemusta suu- remmin häiritse. Nidettä voi suositella kaikille, jotka ovat kiinnostuneet Suomen digita- lisoitumisen historiasta ja No- kian nousun ajasta. Toisaalta kirja on kiinnostava myös leik- kimielisen ja uteliaan miehen elämäntarinana.

## Mediaani- ja keskiarvosuodatusten vertailu

**Y**rjö Neuvon tutkimus- ryhmä Tampereella tunnettiin myös mediaanikoulukunnan nimellä. Kokeillaan esimerkkinä mediaani- ja keskiarvosuodat- timia oheiseen harmaasävy- kuvaan (kuva 1). Uusi kuva saadaan tarkastelemalla jokaiselle pikselille sen lä- hiympäristöä, tässä esimer- kissä 5×5 pikselin kokoisen neliömuotoisen ”ikkunan” sisältöä, ja ottamalla niiden intensiteeteistä joko mediaani (suuruusjärjestyksessä keskimäinen luku) tai kes- kiarvo. Kuvan ulkopuolisten pikselien intensiteetiksi on tässä oletettu aina nolla, mutta parempiakin tapoja on. Erojen korostamiseksi kumpaakin suodatinta on käytetty kymmenen kertaa peräkkäin. Tulokset näkyvät kuvissa 2 ja 3. 📷



Kuva 1. Alkuperäinen harmaasävykuva.

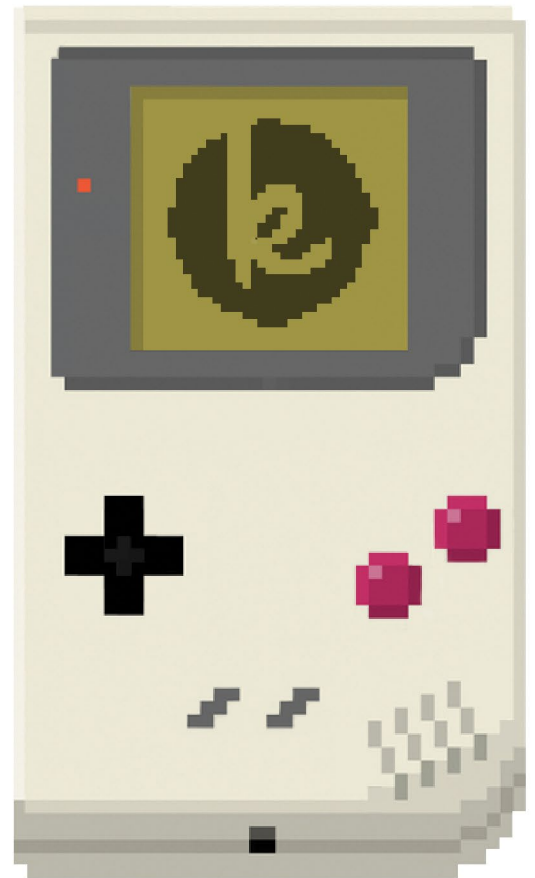


Kuva 2. Mediaanisuo-datettu. Tekstuurit ovat hävinneet, mutta reunat ovat teräviä.



Kuva 3. Keskiarvosuo-datettu. Kuva on kaikin puolin sumea.

# Mukana siellä missä sinäkin.



## Ohjelmointiharjoitus

Toteuta itse mediaanisuo-datatin harmaasävykuville. Suoraviivaisin tapa on käyttää jotakin järjestysalgoritmia erikseen jokaiselle pikselille, mutta tässä on paljon optimoinnin varaa.



Kapsi Internet-käyttäjät ry

# kapsi.fi



# blender™

## Joka paikan 3D-höylä Osa 1, historiasta nykyhetkeen

*Blender on CAD-ohjelmisto, joka soveltuu lähes kaikkeen 3D-grafiikan tuottamiseen. Se sopii niin graafikoille, hakke-reille, muotoilijoille, arkkitehdeille kuin koodaajillekin.*

Teksti ja kuvat: Spiikki Sillanpää

Kuvat: Blender Foundation,

[www.blender.org](http://www.blender.org)

**B**lender on todellinen monitoimiohjelma. Sillä voi esimerkiksi luoda 3D-malleja, piirtää ja liittää niihin pintakuvioita ja jälkikäsitellä laskettuja kuvia. Sillä voi myös tehdä animaatioita tai videoita, jotka koostuvat kuvista, videokuvasta ja ääniraidoista. Jopa pelien tekeminen on mahdollista, ja tulevaisuudessa 3D-mallien tulostuskin voi onnistua.

Nykyään Blender kuuluu useiden graafisen alan yritysten ja peliyritysten työkalupakkiin, ja miksipä ei kuuluisi, sillä työkalu on täysin ilmainen.

### Tie avoimeksi ohjelmistoksi

Vuonna 1988 Ton Roosendaal perusti hollantilaisen Neogeo-animaatiostudion. Seuraavana vuonna hän loi Amigalle Traces-kuvanlaskentatyökalun. Vuonna



Ton Roosendaal, Blenderin isä.

1995 yritys alkoi yhdistellä ja uudistaa käyttämiään työkaluja, ja työn tuloksen opimme myöhemmin tuntemaan Blenderinä.

Roosendaal perusti vuonna 1998 NaN-nimisen (Not a Number) yrityksen, jonka oli määrä myydä Blenderin ympärille tehtyjä kaupallisia tuotteita ja palveluita. Vuosituhannen vaihteessa yritys hankki taakseen myös sijoitusyhtiötä kasvun takeeksi. Valitettavasti surkea kaupallinen menestys ja markkinoiden ankeat näkymät ajoivat sijoittajat lopettamaan yhtiön toiminnan jo vuonna 2002. Tämä tarkoitti myös Blenderin kehityksen pysähtymistä.

Ohjelman ympärille kasvanut yhteisö ja Roosendaal eivät tähän tyytyneet. Toukokuussa 2002 syntyi Blender-säätiö, jonka tarkoituksena oli turvata ohjelmiston kehitys ja markkinointi avoimen lähdekoodin ohjelmistona. NaN-yrityksen sijoittajat hyväksyivät Free Blender -nimisen rahankeräyksen, jonka tavoitteena oli kerätä 100 000 euroa Blenderin oikeuksien ostamiseksi. Monien yllätykseksi rahat saatiin kasaan jo seitsemässä viikossa, ja hankittujen oikeuksien turvin ohjelma voitiin julkaista vapaana avoimen lähdekoodin ohjelmistona.

Uusi vapaa ja ilmainen Blender julkaistiin 13. lokakuuta 2002. Ohjelmistolisenssinä oli GNU General Public License, joka on hyvin yleinen vapaiden ohjelmistojen lisenssi. Samalla Roosendaal sai tukeen suuren joukon innokkaita vapaaehtoisia jatkamaan Blenderin kehitystä.

### Blenderistä tulee tehokas työkalu

Koska Blenderiä oli aiemmin kehitetty Neogeo-studion sisäisenä työkaluna ja omiin tarpeisiin, oli julkiseen käyttöön tarkoitettuna ohjelmaversioon ylläpito alus-hankalaa. Blender-säätiö tarvitsi rahoitusta kehittäjien palkkaamiseen.

Päätettiin aloittaa maailman ensimmäinen avoimen elokuvan projekti, joka toisi ohjelmistolle ja säätiölle näkyvyyttä ja sen myötä rahoittajia. Projektiin koottiin joukko taiteilijoita, jotka haastettiin tekemään innostavaa lyhytelokuvaa.

Hankkeen nimi oli Project Orange, ja sen tuloksena syntyi lyhytelokuva Elephants Dream. Hanke toteutettiin kokonaisuudessaan avoimen lähdekoodin ohjelmistoilla, minkä vuoksi elokuvan kaikki lähdemateriaali pystyttiin julkaisemaan Creative Commons -lisenssillä.

Lyhytelokuvaprojekti oli menestys, ja suosion saattelemana perustettiin kesällä 2007 Blender-instituutti, joka tukee säätiötä sekä yhteisön jäsenten avoimia projekteja. Nykyään instituutti toimii vakituisena toimistona ja studiona.

Vuonna 2008 oli vuorossa Blenderin uudistaminen. Haluttiin uusia ohjelman käyttöliittymä, työkalut, tiedon ja toimintojen käsittely sekä animaatiojärjestelmä. Samalla Blenderin vanhat, 90-luvulta peräisin olevat ydinosat päätettiin ohjelmoida uudestaan. Lähes kaikki tavoitteet saavutettiin vuonna 2011, jolloin Blenderistä julkaistiin versio 2.5.

Vuodesta 2012 eteenpäin ohjelman kehitys on keskittynyt visuaalisten efektien prosessin kehittämiseen. Tähän vaikutti merkittävästi avoin elokuvaprojekti Tears of Steel, jonka myötä ohjelmistoon lisättiin uusi kuvanlaskentamoottori Cycles sekä useita animointia helpottavia työkaluja. Samana vuonna avattiin ammattilaisille suunnattu Blender Network -palvelu, josta saa tukea, koulutusta ja apua ohjelmistokehitykseen. Nykyään Blender-säätiö työllistää kaksi täysipäiväistä ja kaksi osa-aikaista työntekijää.



#### Seuraavassa numerossa...

Seuraavassa Skrollissa tutkimme tarkemmin Blenderin ominaisuuksia ja mahdollisuuksia. Sinun ei kuitenkaan tarvitse odottaa seuraavaan numeroon saakka, sillä voit sukeltaa jo nyt Blenderin ihmeelliseen maailmaan. Artikkelini nimittäin jatkuu Skrollin nettisivuilla, jossa viritämme Blenderin työkuuntoon ja teemme Skrollin logosta 3D-mallin.

**Nettijatko:** <http://www.skrolli.fi/2014.1/>

**Lisätietoja:** <http://www.blender.org>

## Blenderillä tehtyjä elokuvia

### Open Movie -projektit

Blender-ohjelmistolla on tehty useita avoimia, Open Movie -elokuvia.  
<http://www.blender.org/features/projects/>.



#### 2006 - Elephants Dream (Project Orange)

Ensimmäinen Open Movie -projekti.  
[www.elephantsdream.org](http://www.elephantsdream.org)



#### 2008 - Big Buck Bunny (Project Peach)

Ensimmäinen Blender-instituutissa valmistunut avoin elokuva. Projektissa tähdättiin erityisesti partikkeliefektien kuten hiusten ja ruohon parannuksiin. Tästä lähti liikkeelle myös ensimmäinen instituutin kehittämä peli, Yo Frankie. Se toteutettiin kokonaisuudessaan Blenderin sisältämällä Blender Game Enginellä.

<http://www.bigbuckbunny.org> <http://www.yofrankie.org>



#### 2010 - Sintel (Project Durian)

Tavanomaista suuremmalla osallistujamäärällä toteutettu hanke, joka pyrki parantamaan Blender 2.5:n ominaisuuksia, erityisesti uutta sculpt-ominaisuutta. Tulokset julkaistiin Netherlands Film Festival -tapahtumassa. <http://www.sintel.org>



#### 2012 - Tears of Steel (Project Mango)

Projektissa parannettiin jälkikäsitteilyä ja greenscreenin toimintaa sekä uutta realistista Cycles-kuvanlaskentakoneistoa. Mukaan kutsuttiin myös yhdysvaltalainen ohjaaja Ian Hubert. Projekti sai tukea joukkorahoituksen ja The Netherlands Film Fund -rahaston kautta, joten viikon ajan siihen voitiin kuvata materiaalia ammattimaisesti. Cinagrid.nl:n rahoitus mahdollisti elokuvan kuvanlaskennan 4K-tasoisena vuonna 2013. <http://tearsofsteel.org/>

#### 2014 - (Project Gooseberry)

28. tammikuuta julkaistu projekti seuraavaksi lyhytelokuvaksi. Käy tutustumassa! <http://gooseberry.blender.org/>

## Yhteisöt

Blenderin avoimuuden vuoksi yhteisö on kasvaessaankin pysynyt tiukasti ohjelmiston tukena. Youtube on pullollaan ajan tasalla olevia esittelyjä ja oppitunteja ohjelman ominaisuuksista ja tekniikoista. Suurimpia ja aktiivisimpia yhteisöjä ovat Blender Artists, Blender Nation, Blender Cookie sekä Facebookin Blender-ryhmä.

<http://blenderartists.org>

<http://www.blendernation.com/>

<http://cgcookie.com/blender/>

<http://www.facebook.com/groups/2207257375>

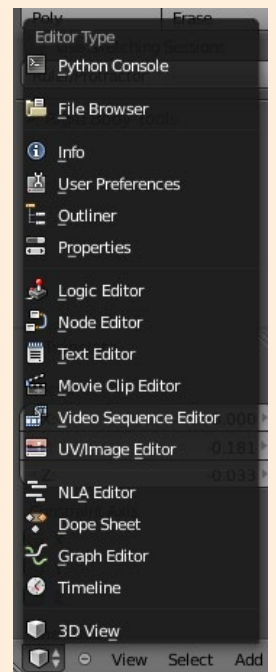
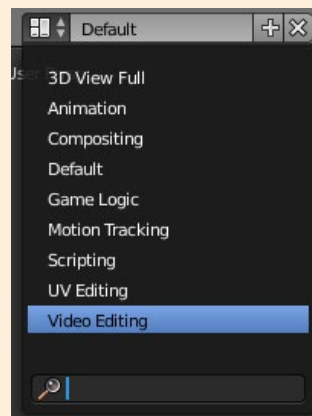
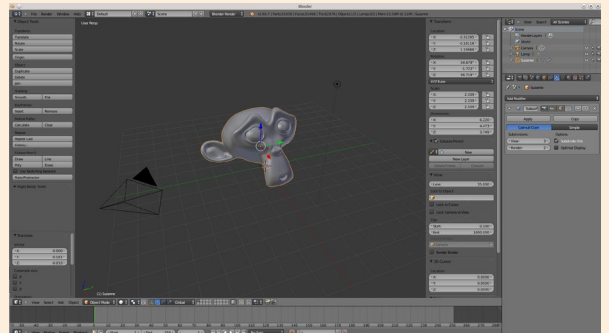
Tietoa yhteisöistä:

<http://www.blender.org/get-involved/user-community/>

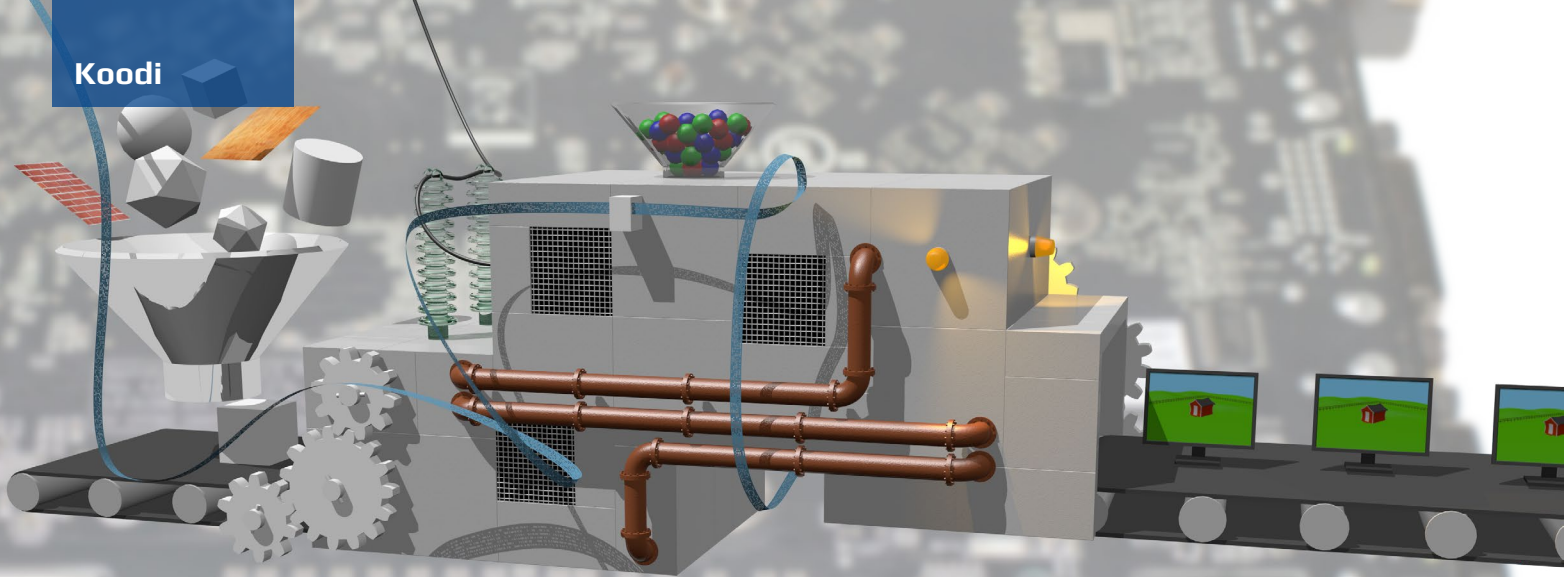
## Blender työkaluna

Blender, kuten moni muukin 3D-ohjelmisto, voi tuntua aluksi sekavalta ja monimutkaiselta. Tätä on vaikeaa välttää, koska ohjelmassa on valtavasti ominaisuuksia ja kaikkiin täytyy päästä käsiksi. Useimmilla valikkotoiminnoilla on omat näppäinikotimensä, joten niitäkin on paljon.

Blenderin käyttöliittymä on rakennettu muovautumaan kunkin käyttäjän tarpeisiin. Käyttöliittymän pystyy muokkaamaan työtehtävän mukaan juuri sellaiseksi kuin käyttäjä haluaa.



Käyttöliittymä koostuu ikkunoista, joita voi vapaasti lisätä ja poistaa sekä asettaa haluamansa kokoisiksi. Ikkunoihin voi määritellä haluamansa muokkaustoiminnot, joita on valittavana 17 erilaista. Ensimmäisellä käynnistyskerralla Blender tarjoaa yleisimpiin tehtäviin yhdeksän valmista muokkausikkunaa.



# OpenGL-ohjelmointi: grafiikkamoottori

*Sarjan kolmessa edellisessä osassa olemme esitelleet OpenGL:n tärkeimmät ominaisuudet. Nyt niputamme kaiken yhteen ja rakennamme yksinkertaisen grafiikkamoottorin.*

Teksti: Mikko Rasa Kuvat: Mikko Rasa, Teija Tuhkio

**A**lkuun varoituksen sana: laadukkaan grafiikkamoottorin tekeminen on valtava urakka. Älä siis odota tekeväsi seuraavaa Unreal Engineä pelkästään tämän artikkelin ohjeilla. Oman grafiikkamoottorin tekeminen on silti opettavaista ja auttaa myös ymmärtämään valmiiden moottorien toimintaa paremmin.

Grafiikkamoottori koostuu useista alijärjestelmistä, jotka tekevät yhteistyötä keskenään. Tärkeimmät niistä ovat resurssien lataus ja hallinta, näkymäverkon (engl. scene graph) hallinta ja renderöinti. Monesti grafiikkamoottorit tarjoavat myös animointitoimintoja. Laajemmissa pelimoottoreissa on myös muita toimintoja kuten törmäystarkistus, käyttäjän syötteen käsittely ja skriptiohjelmat. Tässä artikkelissa keskitymme kuitenkin grafiikkaan.

Grafiikkamoottorin voi rakentaa monella tavalla, eikä tässä artikkelissa ole tilaa esitellä niitä kaikkia. Jos jokin ratkaisu ei miellytä, kokeile rohkeasti toista lähestymistapaa.

Artikkelin mukana tuleva esimerkkikoodi on toimiva grafiikkamoottori mutta ei toteuta kaikkia tässä esiteltyjä ominaisuuksia. Puuttuvien asioiden kohdalla on artikkelissa Puuttuu-merkintä. Lukija voi halutessaan etsiä näistä aiheista lisätietoa internetistä ja kehittää moottoria eteenpäin.

## Yleinen arkkitehtuuri

Allekirjoittanut ei useinkaan tee tarkkoja suunnitelmia etukäteen, mutta jonkinlainen yleiskuva on hyvä muodostaa. Ku-

ten aiemmissakin esimerkkiohjelmissa, moottorin toteutuskielenä on C++. Olio-ohjelmointi on myös luonteva valinta.

Globaalit muuttujat ja muut globaalit tilat voivat tehdä ohjelman kulun seuraamisesta ja virheiden etsimisestä vaikeaa, joten niitä on syytä välttää. Tämän huomaa erityisen hyvin juuri OpenGL:n kanssa, jossa vaikkapa tekstuurin unohtaminen sidotuksi jossain päin koodia voi aiheuttaa virheellistä toimintaa aivan muualla. Niinpä pilotamme OpenGL-toiminnot kokonaisuudessaan moottorin rajapinnan taakse, jossa tiloja voidaan muuttaa hallitusti.

Koska ohjelmat käyttävät grafiikkamoottoria kirjastona, on moottorin luokat ja funktiot syytä nimetä niin, että vältetään yhteentörmäyksiltä toisten kirjastojen kanssa.

Käytön helpottamiseksi teemme moottorille pääluokan, joka tekee kaiken tarvittavan alustuksen. Lisäksi sinne voidaan sijoittaa tietyt ylätasoinnot, kuten kokonaisen ruudun renderöinti. Vältämme kuitenkin muiden luokkien tarpeetonta sitomista pääluokkaan, jolloin niiden uudelleenkäytettävyys paranee.

C++:n muistinhallinnan väitetään olevan vaikeaa, mutta sen ei tarvitse olla sitä. Kun heti projektin alussa päättää pelisäännöt muistinkäytölle, välttyy useimmilta ongelmilta. Skrolli-moottorin esimerkkitoiteutuksessa käytämme omistajuusperiaatetta, eli olion luonut luokka on vastuussa myös sen tuhoamisesta. Käytämme myös pinosta varattuja olioita, aina kun se on järkevää.

Virhetilanteiden raportointiin käy-

tämme poikkeuksia. Muihin virheenkäsitteilytapoihin verrattuna niiden etuna on helppokäyttöisyys. Käsittelemättömät poikkeukset pysäyttävät ohjelman suorituksen, eivätkä virheet jää huomaamatta. Myös poikkeuksen kaappaaminen debuggerissa on helppoa. Moderneilla kääntäjillä poikkeuksien käyttö ei tee koodista hitaampaa, joten ainoaksi haittapuoleksi jää hieman suurempi ohjelman koko.

## Resurssienhallinta

Resurssienhallinnan tehtävänä on ladata ohjelman tarvitsemia resursseja levyltä ja pitää kirjaa jo ladatuista resursseista. Jos samaa resurssia pyydetään uudestaan, resurssienhallinta voi antaa valmiiksi ladatun esiintymän.

Resurssiin on pystyttävä viittamaan jotenkin, joten annamme niille nimet. Suoraviivaisinta on käyttää tiedostonimiä. Resurssija on myös helppointa käsitellä työstövaiheessa erillisinä tiedostoina. Myöhemmin resurssienhallintaan voidaan lisätä vaihtoehtoisia lataustapoja, kuten vaikkapa tuki zip-tiedostoille.

Tärkeimmät resurssityypit ovat objekti, shader-ohjelma ja tekstuuri. Lisäksi tarvitaan joitakin aputyyppejä, jotka eivät kuvaa mitään OpenGL:n olioita vaan auttavat muiden resurssien hallinnassa.

Tekstuurit ovat yksinkertaisesti kuvatiedostoja. Yleisissä kuvaformaateissa ei ole mahdollista helposti määritellä lisäominaisuuksia kuten suodatusta tai toistoja, joten käytämme kaikille tekstuureille samoja asetuksia.

Shaderit ladataan GLSL-kielisestä lähdekoodista. Koska kokonaiseen sha-



der-ohjelmaan tarvitaan sekä kulmapiste- että pikselishaderi, nämä on erotettava tiedostossa jotenkin. Valitsemme erottimeksi vähintään kolme yhdyserkkiä (-) sisältävän rivin, koska se on visuaalisesti selvä erotin eikä sellaista voi normaalisti esiintyä GLSL-koodissa.

Objektit ovat kaikkein monimutkaisin tapaus. Ei ole olemassa mitään universaalia kaikkien käyttämää formaattia, mutta jotkin formaatit ovat levinneet varsin laajalle. Mainitsemisen arvoisia ovat ainakin Collada, 3DS ja OBJ.

Collada on suhteellisen uusi, avoimeen standardiin perustuva XML-pohjainen formaatti, jonka on julkaissut Khronos Group. Teoriassa se voi sisältää mitä tahansa 3D-grafiikassa tarvittavaa dataa. Yleiskäyttöisyytensä vuoksi se on kuitenkin hieman raskas tämän artikkelin tarpeisiin.

3DS puolestaan on Autodeskin 3D Studio käyttämä formaatti. Siitä ei ole julkaistu virallista määrittelyä, mutta sen rakenne on selvitetty perin pohjin. Myös 3DS pystyy säilömään monentyyppistä dataa, mutta binääriformaatin käsittely on työlästä ja altista virheille.

OBJ on peräisin Wavefront Technologiesin mallinnusohjelmista. Siitäkään ei ole kehittäjän julkaisemaa määrittelyä, mutta tekstipohjaisena formaattina se on erittäin helppolukuinen. Yksinkertaisimmillaan yksi tiedosto sisältää yhden objektin, joten se soveltuu mainiosti grafiikkamoottorimme ensimmäiseksi objekti-formaatiksi.

Tilan säästämiseksi en selitä tässä formaattia kokonaisuudessaan. Halukkaat voivat tutustua Wikipedian kuvaukseen tai esimerkiohjelman lähdekoodiin. Pari asiaa on kuitenkin syytä mainita.

OBJ-tiedostossa kulmapisteiden sijainnit, normaalit ja tekstuurikoordinaatit muodostavat kukin oman taulukkonsa. Tasopintojen määrittelyt viittaavat taulukoihin erikseen, jolloin esimerkiksi samaa normaalia voi käyttää usean eri kulmapisteen kanssa. Tämä sopii huonosti OpenGL:ään, jossa kunkin kulmapisteen on oltava oma kokonaisuutensa kaikkine ominaisuuksineen. Niinpä joudumme hieman kikkailemaan saadaksemme datan sopivaan muotoon.

Toinen huomionarvoinen seikka on objektin koostuminen yksittäisistä tasopinnoista. Ne voivat olla myös kolmioita, nelikulmioita tai jopa useampikulmaisia kuvioita. Tämä ei ole kovinkaan tehokasta, koska indeksejä tarvitaan enemmän ja vä-

limuistin tehokkuus kärsii. Optimaalisen ketjutuksen muodostaminen on varsin monimutkaista, mutta onneksi yksinkertaisellakin algoritmilla päästään tyydyttävään lopputulokseen.

3D-mallinnusohjelmissa ei yleensä ole mahdollisuuksia shaderien määrittelyyn, eikä OBJ-formaattikaan tue niitä. Sen sijaan materiaalit ovat yleisesti käytetty ominaisuus, joten käytämme niitä myös Skrolli-moottorissa. Materiaali sisältää vähintäänkin viittauksen shader-ohjelmaan ja mahdollisesti myös tekstuuriin. Lisäksi se voi sisältää shaderin kanssa käytettäviä uniform-muuttujien arvoja.

Materiaaleille ei ole tarjolla mitään yleisesti käytettyä ja yksinkertaista tiedostoformaattia, joka täyttäisi kaikki tarpeemme, joten joudumme kehittämään oman.

Resurssienhallinta omistaa lataamansa resurssit ja tuhoaa ne viimeistään sitten, kun se itsekään tuhoutuu. Julkisen rajapinnan kautta siltä voi pyytää viittauksen tietyn nimiseen resurssiin. Funktiomallineilla voimme myös tarkistaa, että resurssi on odotettua tyyppiä.

Paljon dataa sisältävissä peleissä voi olla tarpeen myös poistaa tarpeettomaksi käyneitä resursseja muistista ja tehdä tilaa uusille, esimerkiksi siirryttäessä kentästä toiseen. Tällöin on huolehditava siitä, ettei mikään toinen resurssi enää viittaa muistista poistettuihin resursseihin. (Puuttuu.)

## Näkymäverkko

Näkymäverkko on tietorakenne, joka kuvaa ruudulle piirrettäviä esineitä ja niiden suhteita toisiinsa. Jotkin grafiikka-

moottorit tuovat sen hyvin vahvasti esiin abstraktion kautta, toisissa se taas muodostuu hyvinkin huomaamattomasti. Skrolli-moottorissa käytämme suhteellisen yksinkertaista ja selkeää verkkorakennetta.

Verkko koostuu render-

deröitävistä asioista, joten luomme abstraktin kantaluokan nimeltä `Renderable`. Eri operaatioista teemme virtuaalifunktioita.

Eräs tärkeä ominaisuus näkymäverkkossa on asioiden ryhmittely. Vähintäänkin tarvitaan yksi ryhmä, joka sisältää kaikki piirrettävät esineet. Yhtä helppoa on kuitenkin tehdä yleispätevä ryhmittelymalli, jolla voi luoda näkymästä hierarkkisen esityksen. Tämä helpottaa asioiden käsittelyä, kun kokonaista ryhmää voi käsitellä yhtenä yksikkönä.

Usein on tarpeen sisällyttää näkymään useampi kopio samasta asiasta. Esimerkiksi metsän jokaisesta puusta on turha tehdä yksilöllistä, koska ne voisivat liikaa muistia eikä katsoja kuitenkaan huomaisi kaikkia eroja. Muutaman erilaisen puumallin monistaminen riittää luomaan vaikutelman elävästä metsästä. Tätä varten teemme `Instance`-nimisen apuluokan, joka pitää sisällään käytettävän matriisin sekä viittauksen alkuperäiseen asiaan.

## Renderöinti

Ensimmäinen ja tärkein näkymäverkkoon toteutettava operaatio on näkymän renderöinti. Siihen tarvitaan jonkin verran globaalia tilaa, kuten kameran ja valaistuksen tiedot. Koska haluamme pitää moottorin eri luokat mahdollisimman itsenäisinä, välitämme nämä tiedot render-funktion parametrina aputietorakenteessa.

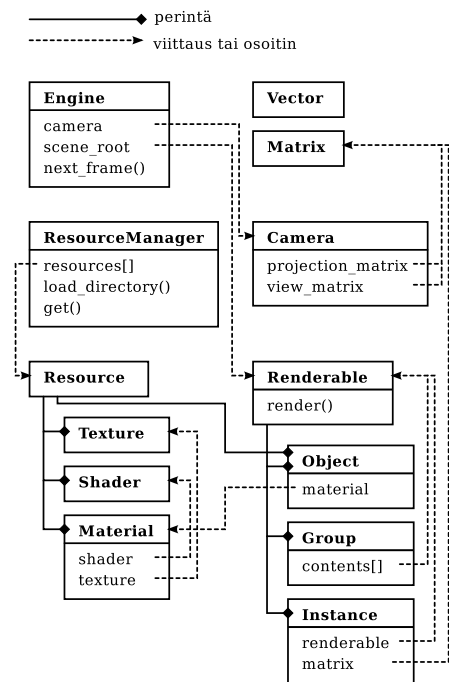
OpenGL:n tilan muuttaminen vie hieman resursseja, joten turhia tilamuutoksia on syytä välttää. Ensimmäinen askel tähän on jättää asetetut tilat voimaan kunkin esineen renderöinnin jälkeen.

Jos seuraava esine käyttää samoja tiloja, ajuri havaitsee, ettei tilamuutosta todellisuudessa tapahdu, ja jättää operaation suorittamatta. Koska OpenGL:n tila on moottorin hallinnassa, voimme rakentaa renderöintilogiikan siten, ettei tahatonta virheteroimintaa pääse syntymään.

Mikäli halutaan piirtää läpikuultavia esineitä kuten värillistä lasia, on läpikuultavat osat renderöitävä muiden jälkeen ja järjestyksessä kauimmaisesta lähimpään. Muuten voi käydä niin, että läpikuultavan esineen takana oleva toinen esine ei piirrykään ruudulle, koska syvyysspuskurissa on jo lähempänä oleva arvo. (Puuttuu.)

Virtuaalimaailman kasvaessa laajaksi on tavallista, että maailmasta on näkyvissä kerrallaan vain pieni osa. Kaikkia esineitä koskevien piirtokomentojen





Luokkakaavio.

lähettäminen näytönohjaimelle on tällöin turhan hidasta. Avuksi tulee tekniikka, jossa etukäteen karsitaan piirrettävien asioiden joukosta pois ne, jotka ovat kokonaan näkökentän ulkopuolella. Hierarkkinen näköverkko auttaa tässä, sillä jos kokonainen ryhmä jää näkökentän ulkopuolelle, ei ryhmän sisältämiä asioita tarvitse käsitellä erikseen. (Puuttuu.)

## Animaatiot

Useimmissa grafiikkamoottoreissa on tuki myös animaatioille. Yksinkertaisimmillaan on kyse esineiden liikuttelusta. Fysiikkaa sisältävissä peleissä pelimaail-

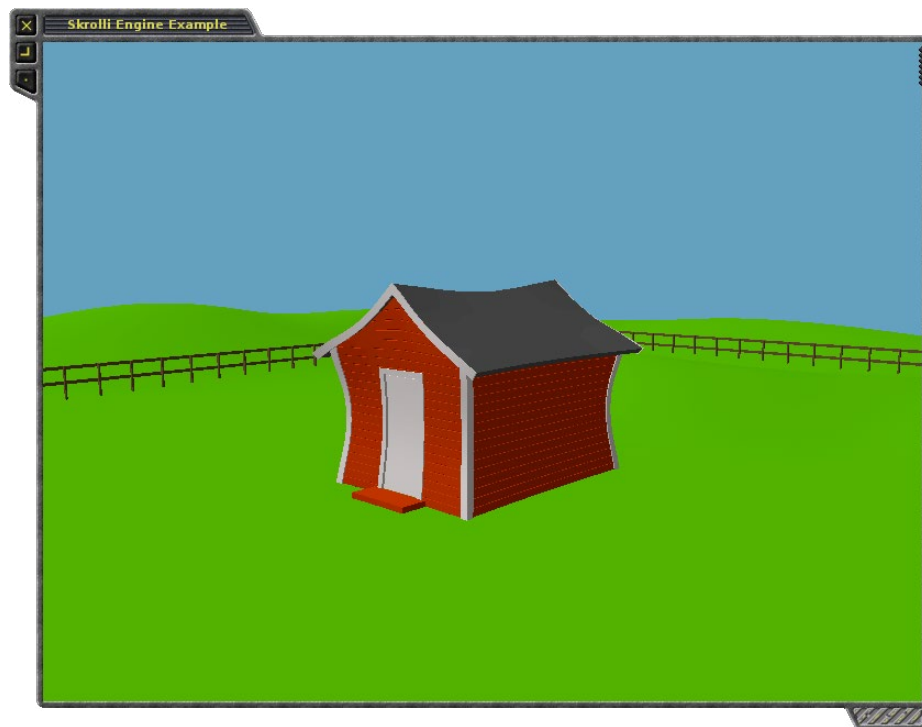
man sisäiset animaatiot lasketaan yleensä fysiikkamoottorilla, mutta se ei kuulu tämän artikkelin aihepiiriin. Peleissä voidaan tarvita myös yksinkertaisempia animaatioita, esimerkiksi käyttöliittymässä.

Jos esine lähtee yhtäkkiä liikkeelle täydellä nopeudella ja pysähtyy kuin seinään, liike näyttää luonnottomalta. Päätepiteitä voi pehmentää sopivalla interpolaatiofunktiolla. Tähän käy esimerkiksi kuutiollinen polynomi  $3x^2 - 2x^3$ . Pidemmät liikkeet voi olla tarpeen jakaa kiihdytykseen, tasaiseen liikkeeseen ja hidastukseen.

Liikkeen ei tarvitse olla suoraviivaista. Useasta viivasegmentistä koostuva murtoviiva on seuraava luonnollinen askel, mutta suunnan äkillinen muuttuminen ei näytä hyvältä. Pehmeiden liikera-tojen määrittelyssä hyvin yleinen työkalu ovat Bezier-käyrät (engl. Bezier curve, Bezier spline). Ne ovat ikään kuin murtoviivoja, joissa jokaiselle pisteelle on annettu sijainnin lisäksi tangentti. Segmentit on väännetty mutkalle niin, että niiden päät sopivat tangentteihin. Bezier-käyrien määrittely ja hienosäätö on käsin työlästä, joten apuna kannattaa käyttää 3D-mallinusuohjelmaa. (Puuttuu.)

Animaatioon voi sisältyä myös pyörivää liikettä, tai se voi koostua pelkästään siitä. Tässä kannattaa käyttää kvaternioita, koska niiden interpolointiominaisuudet ovat ylivoimaiset. Matriiseillakin homma onnistuu, mutta interpolaatio tuottaa vääristymiä, jotka on erikseen kompensoitava.

Elävien olentojen animointiin käytetään usein luurankoja (engl. skeletal



Kuvakaappaus esimerkkiohjelmasta.

```

using namespace SkrolliGL;

Engine engine;
ResourceManager res_mgr;

res_mgr.load_directory("data");

Camera camera;
engine.set_camera(&camera);
camera.set_position(10, -11, 4);
camera.look_at(0, 0, 2);
camera.set_depth_range(0.1, 100);

engine.set_sky_color(0.397, 0.625, 0.748);
engine.set_sun_direction(-0.8, -0.7, 1.3);

Group scene;
engine.set_scene_root(&scene);

Instance ground(res_mgr.get<Object>("ground.obj"));
scene.add(ground);
Instance house(res_mgr.get<Object>("house.obj"));
house.set_matrix(Matrix::translation(2, 0, 0));
scene.add(house);

while(engine.next_frame());

```

Listaus 1.

animation). Mallille tehdään virtuaalinen luuranko ja kulmapisteet yhdistetään yhteen tai useampaan luuhun. Jokaisella luulla on oma matriisinsa, jota muuntelemalla saadaan mallin muoto muuttumaan. Erilaisilla rajoitteilla voidaan varmistaa luiden pysyminen luonnollisessa asennossa. (Puuttuu.)

## Esimerkkitoteutus

Esimerkkitoteutus Skrolli-moottorista on osoitteesta <http://www.skrolli.fi/2014.1/>. Se on sellaisenaan toimiva, joskin rajoittunut 3D-grafiikkamoottori, jota voi käyttää myös pohjana oman moottorin kehitykselle. Koodi on lisensoitu Creative Commons CC0 -lisenssillä, joten sitä voi käyttää, muokata ja levittää täysin vapaasti.

Listauksessa 1 on esimerkki siitä, millä moottoria käyttävä ohjelma näyttää.

Kirjoittajan tavoittaa Ircnetistä nimimerkillä tdb kanavilta #skrolli ja #opengl.fi. Vastaan mielelläni OpenGL:ää koskeviin kysymyksiin.

# Pikseligrafiikkaa: C64 multicolor

*Vaikka Commodore 64:lle onkin kehitetty myös edistyneempiä näyttötiloja, on perinteinen moniväritila edelleen monien taiteilijoiden suosiossa. Seuraavassa muutamia poimintoja viime vuosilta.*

Ville-Matias Heikkilä

C64:n moniväritila:

- Näyttöalue jakautuu 40 × 25 soluun, kussakin 4 × 8 pikseliä.
- Kussakin solussa saa käyttää neljää eri väriä, jotka valitaan kiinteästä 16:n värin paletista. Yksi väreistä (yleensä musta) on sama kaikissa soluissa.

• Paletti:



Yazoo: African Night.



Arsenic: Dreamlands Gate.



Oys: Frog, Landscape and a Lot of Clouds.



Yazoo: Graveyard.



Oys: Ladybird.



Mermaid: What's With All The Hate.



# BATMAN

## Pelejä Gothamista

### Lepakkomiehen lennot

*Jon Ritman käynnistää Lepakkomuistinsa ja kaivelee sieltä tarinoita ajalta, jolloin kahdeksan bittiä riitti vielä kaikkeen. Ritman kertoo, miten kaikkien pelijanoisten sankari siirtyi ensimmäisen kerran peliruuduille pieksämään nollia ja ykkösiä.*

Teksti: Juho Pietarinen Kuvat: Mobygames, Juho Pietarinen, Jukka O. Kauppinen

**80**-luvun alussa pelimaailma näytti kovin erilaiselta. Peli- en päähahmoina eivät seikkailleet elokuvista tutut naamat eivätkä sarjakuvien supersankarit vaan pelintekijöiden omat luomukset. Ocean muutti homman kertatehtävillä. Gary Braceyn johdolla firma ryhtyi haalimaan salkkuunsa lisenssejä toisensa perään.

Samaan aikaan toisaalla: Joulukuussa 1984 Jon Ritman seiso pelifirma Crystal Computingin studiolla ja piti käsissään Knight Lore -peliä. Peli julkaistaisiin vasta viiden päivän kuluttua, mutta pelitalon väki vaati Jonia tutustumaan siihen heti. Aiemmin päivällä Jon oli luovuttanut oman jalkapallopelinsä, ZX Spectrumille tehdyn Match Dayn, julkaisijalleen Ocean Softwarelle, joten mikäpä siinä, hänellä oli aikaa.

Jon suunnisti pelin kanssa koleaan Manchesterin läpi naapurissa asuvien pelintekijäystäviensä luo. Kaikki kahdeksan seisovat rivissä odottamassa pelin latautumista ja käynnistymistä.

Ensimmäiset grafiikat ilmestyivät ruutuun. Seurasi pitkä hiljaisuus, ja Jonin mukaan sen rikkoi lopulta ”ääni joka syntyy usean leuan osumisesta yhtä aikaa lattiaan”.

Knight Lore hyödynsi ensimmäisenä seikkailupelinä isometrista grafiikkaa, jossa kaksiulotteisilla grafiikoilla luodaan kolmiulotteinen vaikutelma näyttämällä

kappaleista kolme sivua. Vastaavaa ei ollut ennen nähty, eikä Spectrumin tehoilla sellaisesta osattu edes haaveilla.

”Me halusimme kaikki tehdä saman”, Jon muistelee sitä talvista iltopäivää. ”Minulla ei ollut käsitystäkään, miten Ultimate oli onnistunut siinä. Tosin yksi kavereistani väitti tietävänsä, mutta hän kieltäytyi kertomasta minulle.”

Knight Lore oli graafisesti suuri harppaus ja Jonin silmissä juuri sitä, mitä hän oli halunnut tehdä itse jo pitkään: Disney-animaatio, jota voi itse ohjata.

”Siihen asti suurin osa peleistä oli ollut kaksiulotteisia, lukuun ottamatta vektorigrafiikoita ja nokkelaa mutta taiteellisesti mitäänsanomaton Ant Attackia. Knight Lore näytti yhtä hyvältä kuin mikä tahansa tv- tai elokuva-animaatio, mutta sitä pystyi myös pelaamaan. Ja minä ihailin, miten huolellisesti erityisesti Disney teki elokuvansa”, Jon kertoo.

#### Pyhä jysäys grafiikka, sä oot kaunista!

”Hullu, vaikea, nerokas.” Näillä kolmella sanalla Jon kuvaa ystävänsä Bernie Drummondia.

Jon oli päättänyt tehdä Knight Loren innoittamana isometrisen pelin mutta tiedosti haasteen menevän liuhuen hänen kykyjensä yli. Jon oli raapustellut ruudulle omia yrityksiään ja tehnyt spritet aiempiin peleihinsä, mutta siihen hänen

graafikon uransa saisi jäädä.

”Minä tunsin kaverin nimeltä Bernie Drummond, joka piirsi paljon sarjakuva-hahmoja”, Jon kertoo.

Jon otti Bernieen yhteyttä ja kutsui hänet vierailulle, taka-ajatusten kera. Vierailua valmistellessaan Jon pyöräytti ZX Spectrumille pienen piirto-ohjelman, ja Bernieen saavuttua hän pyysi miestä esittelemään piirtäjän taitojaan bittien valtakunnassa. Bernie väänsi ruudulle pikseligrafiikkaa, ja Jon seurasi vieressä.

Kaksi tuntia myöhemmin Jonin usko oli loppumassa. Bernie tuntui tiputtelevan ruudulle pikseleitä ilman minkäänlaista logiikkaa, ja lopputulos näytti pelkältä sekasotkulta.

Sitten, yllättäen, pikselisekasotkun keskelle ilmestyi silmä ja ”Bernie muuttui hetkessä hullusta näpertäjästä tekijämieheksi”. Silmän ympärille muodostui nopeasti kokonainen pelihahmo. Budweiserjaksi nimitystä otuksesta tuli yksi pelin ensimmäisistä vihollisista.

Jon oli löytänyt graafikkonsa. Kaksikko päätti kokoontua myöhemmin uudelleen ja pyöritellä ideoita tulevan pelinsä aiheeksi.

#### Pyhä jysäys Robin, me ollaan peli!

”Zap! Pow! Kablam!” sanoi oikeuden nyrkki, kun pyölevä 60-luvun Batman, Adam West, jysäytti Jokerin leukaperiä. Se oli myös Jonin ensimmäinen mieliku-

va, kun Bernie ehdotti Batman-pelin tekemistä.

Batman oli tuolloin yhä nuorten silmissä kova hemmo, joka liiteli synkkyyttä kihisevän Gothamiin yllä ja tervehti rosvoja rystysillään. Samaan aikaan Ocean takoi hittejä lisenssipeleillään, joiden seuraan Batman sopisi kuin pettymys Michael Bay -elokuvan jälkitunnelmiin.

Jon tiesi itsekin, että peli tarvitsi jonkin koukun, jotta pelaajat huomaavat sen myyntititskeillä.

Silti hän ei pitänyt Batmanista. Pönnäkkä sankarihahmo oli ollut suosionsa huipulla 20 vuotta aiemmin, mutta nyt elettiin menevää 80-lukua. Kampausten korkeusennätysten kanssa kamppaili vain Whamin listasijoitus. Ei nuorisoa mitään 60-luvun camp-sarjaa muistaisi, järkeili Jon ja ehdotti pelin aiheeksi mieluummin moderneja Marvel-hahmoja, Thoria ja Herkulesta.

Bernie väitti vastaan: 60-luvun Batman-sarja pyöri yhä tv:ssä, ja sarjakuvat olivat tunnettuja. Jon taipui lopulta. Kaksisikko vei ideansa Oceanin David Wardille, joka innostui niin, että ”hän ravasi ympäri toimistoaan kuin junaa matkiva lapsi ja lauloi Batmanin tunnussäveltä”.

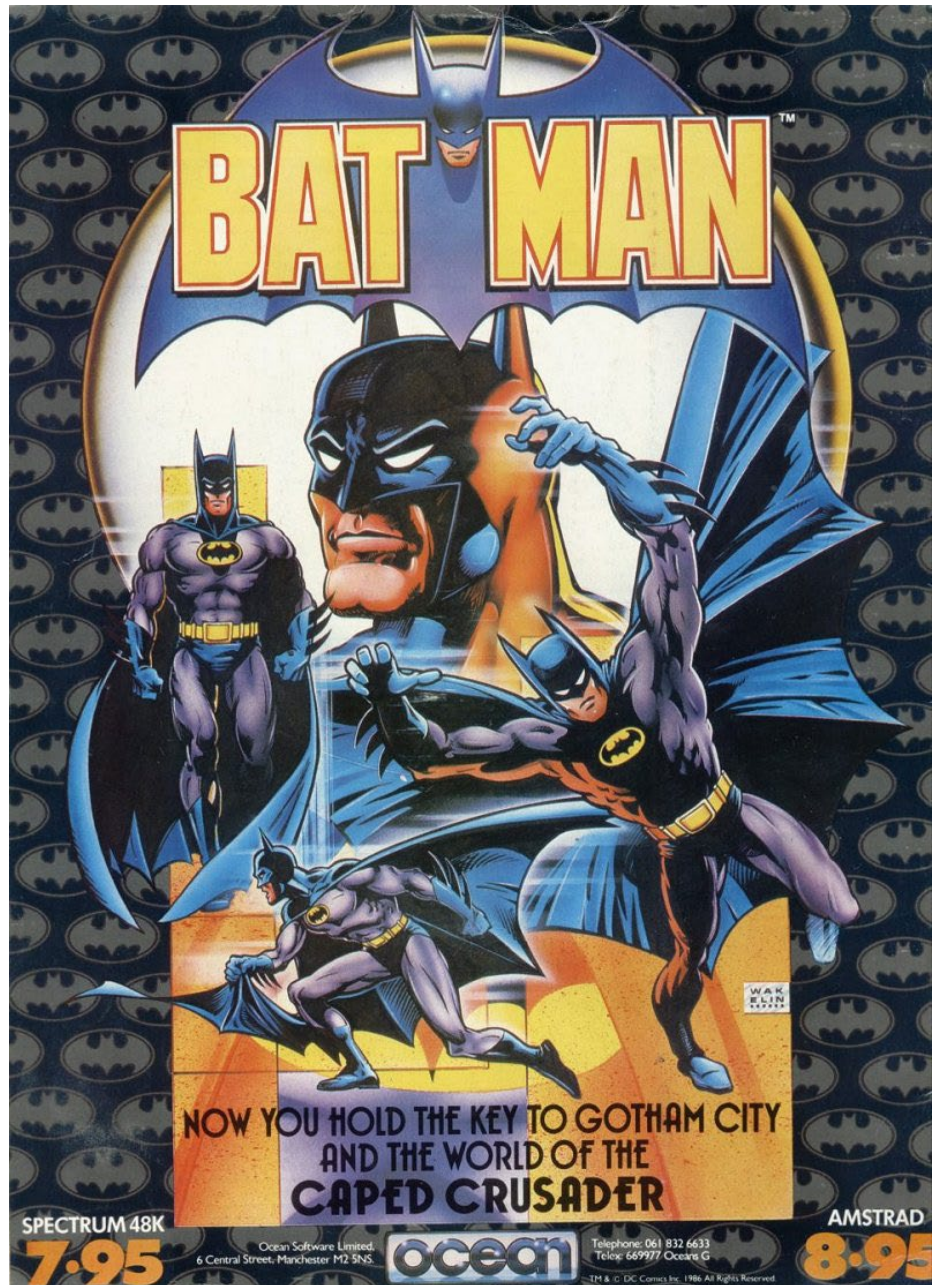
Pari viikkoa myöhemmin Ward ilmoitti, että Ocean oli hankkinut lisenssit DC Comicsilta. Nyt se oli varmaa: Jon Ritman ja Bernie Drummond tulisivat tekemään kaikkien aikojen ensimmäisen Batman-pelin.



### Pyhä jysäys seinä, oot sä Robinia nähnyt?

Jon kulutti seuraavat kaksi kuukautta järkeilemällä, miten Knight Loren Filmation Engine -pelimoottori oli tehty. Pelinteon tärkein väline oli British Micron CP/M-pohjainen tietokone, johon kytkettiin sarjakaapelilla muita koneita, joille peli oli tarkoitus kääntää. Ohjelmointityyli oli aikakautensa näköinen.

”Tapasin tulostaa vanhalla Epsonin matriisitulostimella metrin mittaisia satseja koodia. Se on yksi asia, jota kaipaavan nykyajan koodausympäristössä. Mikään ei voita metrin mittaista tulostetta, jolla voi sitten käsin yhdistellä koodia kaavioksi”, Jon muistelee.



Koodia ja vuokaavioita väännettiin aamupäivästä aamuyöhön. Välissä oli taukoja vain kahville ja tyttöystävälle. Lopulta piti kerätä osaset kasaan ja muovata niistä pelimoottori. Ja näytettävä pers... pitkää nenää sille Manchesterin kaverikämpältä tutulle tyypille, joka ei suostunut paljastamaan Knight Loren salaisuuksia - jos tämä niitä edes tiesi.

Jonin isometrinen pelimoottori oli lopulta jopa parempi kuin esikuvansa.

”Vuosia myöhemmin, juteltuani Knight Loren pelimoottorin ohjelmoijan kanssa, minulle selvisi, että Batmanin pelimoottori oli tehokkaampi. Minulla oli selvästi pienempi graafinen puskurimuis-ti (Batman 256 tavua; Knight Lore 6 kilotavua) ja huonerakenteet tallennettiin erittäin tehokkaasti.”

Samalla pelin suunnitelmat elivät.

Ensimmäisen käsikirjoituksen mukaan Robinin piti vilahtaa seikkailussa

mentorinsa vierellä. Bernie ehti piirtää Robinin spritenkin valmiiksi, mutta ZX Spectrumin muistin rajat tulivat vastaan.

Samalla Your Sinclair -lehti, joka sai käsiinsä lähes valmiin version pelistä, väitti siloposkisen ihmepojan olleen alkuaan jopa pelin pääosassa. Lehden mukaan Jon ei vain onnistunut keksimään, miten voisi ohjata kahta supersankaria yhtä aikaa. Jonin mukaan väite on silkkaa potaskaa.

Jon todistikin väitteen vääräksi jo seuraavassa pelissään, Head Over Heelissä, jossa pelaaja ohjaa kahta päähenkilöä isometrisessä seikkailussa. Batmanin ja Ihmepojan ongelmat ratkesivat kuin itsestään, sillä onnekaasti Robin kidnappattiin. Niinpä yksin jääneen Batmanin täytyikin nyt pelastaa Robin pulasta.

Se ei kuitenkaan onnistuisi, ennen kuin seniilin Alfredin jäljiltä saataisiin kasaan kaikki tehtävän ratkaisemiseen

tarvittavat neljä Lepakkotyökälyä. Näiden avulla maailman parhaan etsivän täytyi vielä etsiä Lepakkoluolan osat, jotka Alfred oli tiettenkin leivittänyt pitkän Lepakkoluolan ansoitettuja ja mörököllien valtaamien huoneita.

Ja luolan sisälle tämä oikeuden puolustajan ensimmäinen seikkailu jäisikin, sillä pelaajaa ei kutsuttu mukaan pelastamaan Robinia.

### Pyhä jysäys Spectrum, tässä on hitonmoinen duuni!

Jon ohjelmoi peliä lähes vuoden. Samalla Bernie toi jatkuvasti uutta grafiikkaa, vaikka Jonin ohjeet olivat tasoa "tätä ja tätä kokoa olevia juttuja, kiitos".

"Minä en varsinaisesti välittänyt grafiikoiden asiayhteydestä", Jon kertoo. "Halusin vain juttuja, jotka näyttivät hyvältä. Tajusin jo aikaisessa vaiheessa, että jos kaikki olisivat samassa mittakaavassa, se rajoittaisi hyvien kuvien määrää rajusti. Lopulta päädyin vain kertomaan Bernielle haluamieni kuvien koon."

Se johti aika mielenkiintoiseen vihollisvalikoimaan.

"Siksi pelissä on Batmania puolet pienempi kaunis teepannu, joka olisi oikeassa mittakaavassa vain kaksi pikseliä leveä ja siis tilanhukkaa", Jon kertoo. "Eri koot saivat lempinimiä: esimerkiksi 'makeiset' oli yhden nimi."

Osa grafiikoista heivattiin pois, osa otettiin käyttöön. Mitä oudommalta grafiikka näytti, sitä parempi. Vihollisiksi päätyivät muun muassa käyskentelevät päät ja kauniit teepannut. Eivät mitenkään tyyppillisimpiä Yön ritarin arkkivihollisia.

Lisenssin omistaja DC Comics oli karsivallinen, vaikka Jon revitteli omalla tyylillään. DC ärähti vasta, kun power upit nimettiin Lepakkopilleriksi: "Ei Batman käytä lääkkeitä!" Jon nimesi pillerit Lepakkovoimiksi, ja kaikki olivat tyytyväisiä.

Batmaniin keksittiin myös aloituskohdista pelihahmon kuoleman varalta. Ne aktivoituivat Batmanin koskettaessa kiveä, joka mahdollisti jälleensyntymän. Sellaisia tuli vastaan silloin tällöin.

Pelinkeskeisyys oli hauskaa mutta myös kovaa työtä, ja projektin mittakaava oli jättimäinen.

"Batman ja Match Day olivat laajuudeltaan lähes samanlaisia ja työhön osallistui samankokoinen työryhmä, mutta aiemmin pelini olivat olleet kolmen kuukauden projektteja. Molemmat olivatkin aiempaan verrattuna valtavia. Batman oli

näistä laajempi, koska minun piti vielä rakentaa kaikki ne huoneet."

Niin. Huoneita huoneen perään. Ja lisää huoneita. Tarkalleen 150, tai 153, tai jotain siltä väliltä. Ehkä. Tarkalleen. Jon ei enää muista eikä muistanut edes silloin, kun peli ilmestyi vuonna 1986. Kukaan ei vaivautunut laskemaan. Jokaisessa huoneessa on oma ongelmansa, omat vihollisensa ja esteensä. Toisissa huoneissa saattaa tapahtua suoraan liukumäkeen, toisissa liukuhihnalle. Muisa saattaa lattiaa kadota alta ja Lepakkomiestä odottavat piikit.

Yhdessäkään ei voinut rentoutua, mutta lähes kaikki oli läpäistävä. Pieninkin kosketus viholliseen sai kuumaverisen Batmanin katoamaan savuna ilmaan. Uudelleenpeluuarvo ei ollut suuri, mutta läpäisy kesti kauan. Onnekkaita olivat ne, jotka loppuun asti pääsivät.

Jon rakensi peliin myös kolme vaihtoehtoista äänimaailmaa: Nasty toisti kaikki äänet, Useful ärähti vain Batmanin törmäillessä ja Late at Night oli yhtä äänetön kuin nörtti cocktail-kutsuilla.

### Pyhä jysäys Ocean, taasko?

Ocean oli luonut mainetta julkaisijana, joka aloitti peliensä mainoskampanjat etuajassa mutta luisui luvatuista julkaisuajankohdista kerta toisensa jälkeen. Batman osoittautui poikkeukseksi. Se tähdättiin alkujaan joulukuksi 1985, mutta Ocean huomasi Jonin olevan eri mieltä.

"Minä kyllä tähtäsin joulukuulle mutta vain ympärilyövästi", Jon kertoo. "Minun asenteeni on, että peli on valmis vasta kun se on valmis, ei silloin kun joku johtaja sanoo, että sen tulisi olla valmis."

Peli julkaistiin pääsiäisenä 1986 ja Ocean aloitti massiivisen mainoskampanjan: Caped Crusader, vihdoinkin pelinä! Ajankohta osoittautui täydelliseksi. Peli-lehdet olivat keväällä nälkäisiä, kun uusia pelejä tuli madellen. Niinpä Batman kaappasi lehtien etusivut. Tulihan se sentään yhdeltä suurimmista julkaisijoista,

ja pelin päähahmona oli yksi legendaarisimmista supersankareista.

Myöhemmin Jon kuuli, että monen lehden graafikot olivat innoissaan päästessään piirtämään Batman-kansia. Aiemmin he olivat raapustelleet Batmaneja vain luonnoslehtiöiden reunoihin.

Batmanin menestys valui myös sisäisille. Your Sinclair pykäsi pisteitä 9/10, Crash puski perään 93 % -arvostelun ja Sinclair User nimesi pelin suoraan klassikoksi viiden tähden arvosanalla.

Myyntiä ei haitannut sekään, että samana keväänä julkaistiin Frank Millerin neliosaisen Batman-sarjakuvan ensimmäinen osa, Yön ritarin paluu, joka osataan nosti kevään lepakkokuumeesta.

Jon, joka oli oppinut ohjelmoimaan ensimmäisen ZX81-koneensa mukana tulleen oppaan avulla, oli luonut toisen peräkkäisen menestyselämänsä.



### Pyhä jysäys, pyhä jysäys, sä toistat itseäsi!

Match Day oli mahdollistanut Jonin siirtymisen täyspäiväiseksi pelinikkariksi, mutta Batman muutti hänen elämänsä, kuten Jonin mukaan "rahalla on tapana tehdä". Samalla se antoi meille pelaajille tärkeimmän asian: uuden pelin, jonka Jon ideoi täysin itse.

Head Over Heelsin ideat olivatkin jo porisemassa, mutta miksei Batman II:ta koskaan nähty?

"Lisenssoitujen projektien ongelmana on lisenssin omistajille menevä raha. Päätelin, että Match Dayn ja Batmanin jälkeen ihmiset tunsivat minut jo niin hyvin, että selviäisin omilla pelihahmoillani", Jon selittää. "Se toteutui Head Over Heelsissä."

Siinä, Jon, olit totisesti oikeassa.



## Meren huuhtoma Jon Ritman

Vuonna 1983 saarivaltakunnan kauppojen ikkunat ja brittinörttien sydämet valtasi C64:n International Soccer -jalkapallopeli. Tai ainakin niiden harvojen, joiden kotoa C64 löytyi, sillä kuningaskunta ei ollut Suomen tapaan tasavallan tietokoneen ominaisinta temmellysaluetta. Bri-



## Ocean ja lisenssipelit

Brittiläinen Ocean ei suinkaan itse keksinyt ideaa menestyselokuvien lisensoinnista videopelimuotoon. Se kunnia kuuluu Atarille, joka loi Atari 2600-konsolille Raiders of the Lost Ark -pelin Steven Spielbergin elokuvan pohjalta. Atari kiirehti hutaisemaan myös surkean lisenssiversion E.T.-elokuvasta, mikä osaltaan auttoi romauttamaan konsolipelimarkkinat vuoden 1984 hujakoilla.

Mutta vasta Ocean teki lisenssipelistä tuottoisan, kestävänsä bisneksen. Yhtiön bisnesjohtaja, leffafriikki Gary Bracey haistoi, että virallisten lisenssipelien tekeminen on tuottoisa bisnes. Hän myös teki siitä Oceanin bisnesmallin ydintä.

tannian markkinoita hallitsi Sinclairin ZX Spectrum, joka yhä oli vailla jalkapallopeliä.

Artic Softwarella tuolloin työskennellyt Jon ryhtyikin laatimaan Spectrumille suoraa kloonina International Soccerista, vaikka jalkapallo ”tylsistyi hänen kuoli-aaksi”.

Kaksi viikkoa myöhemmin Jon seisoilavan edessä Lontoon tietokonemessuilla ja tuijotti Artic Softwaren toisen jalkapallopelin, mitäänanomattoman World Cup Footballin esittelyä ZX Spectrum 48K:lla. Peli ei vakuuttanut.

Jonin vieressä seisoil sattumalta David Ward, joka oli vuotta aiemmin perustanut Jon Woodsin kanssa Ocean Softwaren. Jon mainitsi itsevarmana Davidille, että hänen jalkapallopelistään tulisi paljon parempi kuin tuo.

Tämä sattumanvarainen kohtaaminen unohtui yhdeksäksi kuukaudeksi, ja Jon uppoutui luomaan klooniaan ZX Spectrumille. Kaksi viikkoa ennen pelin valmistumista Jonin puhelin soi.

”Teitkös sinä koskaan sitä jalkapallopeliäsi?” soittaja kysyi.

Soittaja oli David Ward, joka tarjosi Jonin pelistä ennakkona 20 000 puntaa. Se oli kova raha, sillä Jonin vuosipalkka tv-korjaajana oli 7 500 puntaa. Puhelun loputtua Jon oli täysipäiväinen pelinkehittäjä.

Artic Software sai jäädä, vaikka se

### Jon Ritmanin pelihistoria

- Namtir Raiders, Artic Computing Limited, 1981 – Sinclair ZX81
- Cosmic Debris, Artic Computing Limited, 1982 – Sinclair Spectrum 48K
- 3D Combat Zone, Artic Computing Limited, 1982 – Sinclair Spectrum 48K
- Dimension Destructors, Artic Computing Limited, 1983 – Sinclair Spectrum 48K
- Bear Bower, Artic Computing Limited, 1983 – Sinclair Spectrum 48K
- Match Day -pelisarja, Ocean Software, 1984 – Sinclair Spectrum 48K, Commodore 64, Amstrad CPC, BBC Micro
- Batman, Ocean Software, 1985 – Sinclair Spectrum, Amstrad CPC & PCW, MSX, Einstein, Enterprise
- Head Over Heels, Ocean Software, 1987 – Sinclair Spectrum, Amstrad CPC & PCW, MSX, Commodore 64, Atari ST
- Monster Max, Rare/Titus, 1994 – Game Boy
- Super Match Soccer, Cranberry Source/ Acclaim, 1998 – PC, Playstation

olikin julkaissut Jonin ensimmäisen pelin, Namtir Raidersin. Nimessä on muuten tietty logiikka, ja sen bongaa jille on luvassa virtuaalikunniainia.

Tavoilleen uskollisena Ocean yritti ostaa Jonin peliin BBC:n jalkapallo-ohjelma Match of the Dayn lisenssin mutta ei onnistunut. Ovelasti lähes kaimana julkaistu Match Day myi silti yli 50 000 kappaletta. David Ward oli tunnistanut helmen!

tehtiin käsikirjoituksen pohjalta, näkemättä sekuntiakaan valmiista elokuvasta. Kuvausten aikana tehdyt muutokset käsikirjoitukseen saattoivat paljastua pelinkehittäjille vasta jälkikäteen.

Vaikka Ocean kantoikin lisenssipelitalon mainetta, olivat monet yhtiön julkaisuista kiistämättömän erinomaisia pelejä – siitäkkin huolimatta, että osa niistä tehtiin jokseenkin kaavamaisella reseptillä. Toisaalta esimerkiksi Lucas Arts -henkinen seikkailu Hook osoittaa, että yhtiöllä riitti uskallusta myös kaavasta irrottautumiseen.

Oceanin kuuluisimpia lisenssipelilejää olivat muun muassa Cobra, Daley Thompson’s Decathlon, Highlander, Hook, Jurassic Park, Knight Rider, Lethal Weapon, Platoon, Rambo, Red Heat, Street Hawk, Terminator 2, The Addams Family, The Neverending Story, Top Gun, Total Recall, V ja WWF Wrestlemania.

Yhtiötä on käsitelty tuoreeltaan laajasti teoksessa The History of Ocean, jonka voi tilata osoitteesta <http://www.oceanthehistory.co.uk>.

# Puhu mulle konekieltä, beibi

Ei ole sattumaa, että Lepakkomies pelas- teli Robinin persasta nimenomaan Z80- prosessoriin perustuvilla koneilla, sillä Jon pystyi sovittamaan ZX Spectrumille tekemänsä pelin muille Z80-pohjaisille koneille muutamassa päivässä.

”Jokainen versio vaati vain kolmen tiedoston kirjoittamista uudelleen: yhden grafiikoille, yhden äänille ja yhden näppäimistöille”, Jon muistelee. ”Usein selvisin lähes ilman minkäänlaisia muutoksia, joten eri versiot syntyivät parissa päivässä.”

Seuraavassa on lueteltu alustat, joille Lepakkomies liihotti.

## MSX

Lähimpänä alkuperäisversiota. MSX:n äänet olivat paremmat kuin Spectrumin, mutta ominaisuuksien hyödyntäminen jäi noin viiden sekunnin luritukseen aivan pelin alussa ja lopussa.

## Amstrad PCW

”Minulla on hassu muisto siitä, kun taju- sin saavani käännettyä pelin tekstin- käsittelylaitteelle ja huomasin, että laite pystyy pelkkiin piippauksiin”, Jon kertoo. ”Hitot siitä”, hän ajatteli, ”kokeillaan.” Jon lykkäsi PCW:n mölinät keskelle ZX Spectrumin ääniohjainta.

”Olin ällistynyt, kun laite piippasi ulos Batmanin tunnussävelen!”

PCW-versio on yksivärinen eli sairaal- la tavalla joko vihermusta tai piristäväs- ti sairaalavuoteen valkoinen. Mutta hei, Batmania tekstinkäsittelylaitteella, voiko olla parempaa?

## Amstrad CPC

Perheen värileima, kiitos CPC:n laajem- man väripaletin. Peli käyttää täyttä vä- ripalettia, mutta niistä voidaan käyttää kerrallaan vain neljää väriä. Tämän vuok- si kyseessä on monen mielestä nätein versio.

## Enterprise 64/128

Kummajainen pelikoneeksi, kaunotar peliksi. Enterprise 64 ja 128 olivat tie- tokoneina teknisesti yliverkaisia muihin Batman-alustoihin verrattuna, mutta peli vastasi pitkälti Amstrad CPC:n versiota.

Lepakkovararengas: Hieman pulska- na pelissä seikkaileva Batman perustui

ulkomuodoltaan ja animoinniltaan 60-lu- vun camp-henkisen tv-sarjaan, ei DC:n sarjakuvista tuttuun fitness-supersanka- riin.

## Yön witarin uusversiot

Batman on kohentanut rintamustaan ja kiristänyt naamaansa reilun vuosikym- menen välein.

Vuonna 2000 taitonieikka Kak ohjel- moi DOS-alustalle uuden version, Wat- manin. Uusversio on pelillisesti uskollin- en alkuperäiselle, mutta sen grafiikat ja äänet loikkasivat vuosikymmenen verran eteenpäin.

Kymmenen vuotta myöhemmin oli To- mas Kacin vuoro. Ohjelmoituaan ensin oman näkemyksensä Jonin Head Over Heelsistä Tomas pakotti myös Batmanin uudelleen esteradalle. Hänen versionsa käytti samaa pelimootoria mutta tarjosi Lepakkoluolasta varjostetumman versi- on.

Mutta miksi juuri Batman?

”Se oli yksi lapsuuden lempipeleistä- ni”, Tomas kertoo. ”Resoluutio on tuplas- ti alkuperäistä korkeampi ja se pyörii 60 ruutua sekunnissa, joten meidän täytyi tehdä Batmanin liikkumisesta sujuvaa mutta säilyttää ongelmien ratkaisut alku- peräisversiolle uskollisina. Yleensä uusi- en versioiden täytyy olla mahdollisimman identtisiä alkuperäisen kanssa, jotta fiilis pysyy oikeana.”

”Fiilis” on juuri oikea sana peliä ku- vaamaan, sillä täsmälleen sitä Tomas sai peliin jopa lisättyä. Värikkäämpi mutta monin paikoin tummempi grafiikka sekä taustalle ilmestynyt musikaalinen jytinä tuovat peliin uutta eloa.

Entäs Jon, mitäs mies oli mieltä pelis- tä?

”Otin Joniin yhteyttä ja sain hänen siunauksensa”, Tomas kertoo. ”Valitetta- vasti vain peli ei pyörinyt hänen koneel- laan, joten Jon ei pystynyt pelaamaan sitä.” 🦇

## Batman

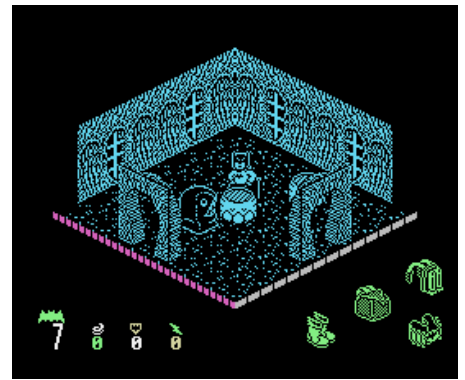
**Julkaisija:** Ocean Software

**Kehittäjä:** Jon Ritman

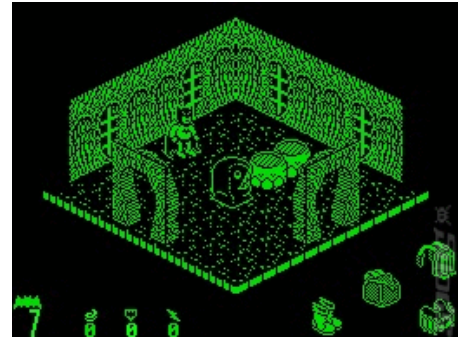
**Julkaistu:** 1986

**Alustat:** Amstrad CPC/PCW, MSX, Enterprise, ZX Spectrum

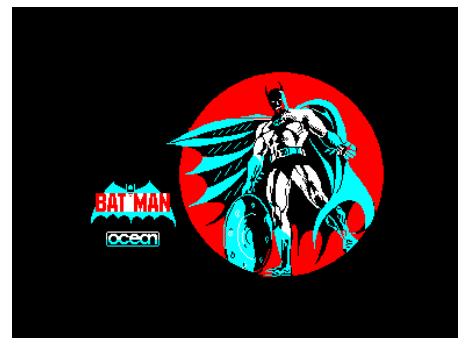
**Lajityyppi:** isometrinen seikkailupeli



MSX.



PCW.



CPC.



Enterprise 64/128.



Batman Remake.

Älä koe Jonin kohtaloa, vaan get yours now, from an internet near you!

- Watman: <http://www.hotud.org/component/content/article?id=22495>
- Batman: <http://retrospec.sgn.net/game/batman>



# Aikaetsivä

## Unholasta palannut seikkailu

Jukka Tapanimäki oli 1980-luvulla Suomen toiseksi kuuluisin pelintekijä. Mutta oliko hänen ensimmäinen kaupallinen pelinsä, tekstiseikkailu Aikaetsivä, oikeasti mistään kotoisin?

Teksti: Ville-Matias Heikkilä

**M**onet luulevat Aikaetsivän kadonneen sen jälkeen, kun Triosoftware maksoi pelistä Tapanimäelle 400 markkaa ja jätti sen julkaisematta. Mutta ei suinkaan! Triosoftware oli jo tehnyt pelistä kopioita, ja suurin osa niistä myytiin vuosien kuluttua asiakkaille tyhjinä disketteinä. Ainakin muutama peli myytiin puolihuolimattomasti vakioasiakkaille. Osa lerpusta päätyi vuonna 1993 Topaz Beerline -ryhmälle, joka teki pelistä piraattijulkaisun. Myöhemmin Aikaetsivä löydettiin uudestaan, kun C64-materiaalia arkistoiva Jarkko "Grue" Lehti sai Jukka Tapanimäen leskeltä tämän vanhat disketit.

Peli alkaa aikalinjan korjaustöihin erikoistuneen firman toimistolta, jonne on juuri tullut viesti: Zorbus-niminen vanki on karannut menneisyyteen, mistä hänet pitäisi noutaa takaisin, jotta maailma pelastuisi. Pelaaja astuu aikasiirtimeensä, samoillee hetken toisessa ajassa ja päätyy sitten hylättyyn tehtaaseen, joka on pelin päämiljö.

Aivan aluksi pelaajan on kuitenkin opeteltava lukemaan. Tapanimäki nimittäin keksi sovittaa vanhat peruskoulun kaunokirjaimet 8×8 pikselin merkeiksi, jotka lähinnä rumentavat peliä ja vaikeuttavat tekstin lukemista. En itse jaksanut totutella koukeroihin vaan korvasin ne selvemmillä. Käyttämäni versio pelistä on ladattavissa myös Skrollin sivuilta.

Suurin osa etsiväntyöstä on esineisiin liittyvien pulmien ratkomista. Kuinka saada liekinheitin toimimaan? Mistä väline pulttien avaamiseen? Työkalupakkia ei ilmeisesti pystynyt ottamaan mukaan aikamatkalle. Muita eläviä olentoja on pelissä hyvin vähän, ja nekin ovat lähinnä kertaluonteisia tekstiseiniä. Mutta mitä vähästä! Onhan hylätyn tehtaan koluminaaminen ja romurakentelu kivaa yksinkin.

Aikaetsivä on selvästi lähinnä tekninen taidonnäyte, pelimoottorin esittelyteos. Eräässä huoneessa eri-

värisiin painikkeisiin on viitattava adjektiiveilla. Toisesta paikasta taas löytyy laatikko, jossa on sisäkkäin todella monta esinettä. Juuri tämän tyyppisiä hienouksia oli ulkomaisissa tekstipeleissä, ja niiden toteuttamiseen paneuduttiin myös suomalaisien lehtien ohjelmointiartikkeleissa.

Tapanimäen tekninen kunnianhimo jättää sisällön pahasti sivuun. Juonesta ei voi juuri puhua. Virheilmoitusten letkauktelemiin vitseihin kyllästyy pian, ja joka toisessa huoneessa on jotain "massiivista". Tarinankerronnallista yritystä löytää vasta loppupuolella, kun pelaaja huikutetaan oman päänsä sisään katselemaan muuttuvia hallusinaatioita.

Vaikuttaa todennäköiseltä, että Aikaetsivä jätettiin julkaisematta siksi, että se oli yksinkertaisesti huono. Mutta mitä jos se olisikin julkaistu? Olisiko suomalaisia tekstipelejä kenties julkaistu sen innoittamana paljonkin enemmän? Olisiko Tapanimäen pelimoottoria käytetty enemmänkin, ja olisiko hänen oma pelijulkaisuhistoriansa muotoutunut kovinkin erilaiseksi? Ehkä saamme kysymyksiin vastaukset, kunhan aikasiirrin joskus keksitään.

## Ei kaupallista potentiaalia

Teksti: Jukka O. Kauppinen

**S**uomenkieliset tekstiseikkailut eivät koskaan yleistyneet, ja käytännöllisesti katsoen genre kuoli lähtökuoppiinsa hyvien suomenkielisten parserien puutteesta. Harrastajien harvat luomukset ymäränsivät vain yksinkertaisia lauserakenteita, ja genren varhaisista edustajista suurin osa onkin kadonnut digihistorian joukkohautoihin.

Työkalujen puutteen ja kielikuilun takia tekstiseikkailujen sijaan



```
>Käynnistä aikasiirrin
Vainaan tapanimäki sinut aika-avaruuden
myrteisiin, jossa alkaa sinulle tuttu
myrkytys komeen yrittäessä säilyttää
kehoasi atomit edes jonkinlaisessa
järjestyksessä siirtymisjaksen aikana.
Et koskaan ole tottunut ajatukseen,
että matkan aikana sinut hajotetaan ja
taas koetaan miljoonia kertoja, Eteen
juuri nyt.
Vähitellen sinua ympäröivä kaaos alkaa
saada kaafon...
YRKKÄ MÄKI
(Olet aikasiirtimen sisällä.)
Olet heikkellä jyrkällä mukaa, joka jatkuu
ylös pohjoiseen ja alas etelään. Tien
molemmiin puoliin näet silmät hallioloita-
kaukset, jotka rajoittavat näkyvyyttä.
Läsnä on huteru puinen vaji.
Aikasiirtimessä on tulla.
>
```

```
>ajattele
Aivosi toimivat hyvin, mutta lukitsin
pitää sinua yhä vankinaan.
>
```

```
>p
Anteeksi kuinka? Yritätkö liikkua oman
pääsi sisällä?? No, olkoon menneeksi,
mutta käytä siinä tapauksessa käskyjä
OTSIA, TAKARAIVO, OIKEA, VASEN, YLÖS ja
ALAS. Nimittäin en tiedä mihin
ilmansuuntaan nenäsi osoittaa.
>
```

```
>otsia
Aistit ympäristössäsi muutoksen. Näet
suuren määrän esineitä. Niiden muoto
muuttuu nopeammin kuin ehdit seurata.
>
```

```
>takaraivo
Aistit ympäristössäsi muutoksen. Näet
kerrostalon korkeisen puoliksi mädän
banaanin.
>
```

Tekstipeleistä voit lukea lisää Skrolli 2013.4:n artikkelista Tekstipelit viideltä vuosikymmeneltä. Suomalaisia C64-tekstipelejä löytää osoitteesta [www.ntrautanen.fi/computers/commodore/archive/c64\\_tekstipelit.htm](http://www.ntrautanen.fi/computers/commodore/archive/c64_tekstipelit.htm).

# Vampire 600 -kiihdytinkortti Pippuria pikku-Amigaan

*Commodore Amiga on eräs suosituimmista retroharrastajien virittelykohteista. Uusi Vampire 600 -kortti antaa roiman piristysruiskeen Amiga 600:lle.*

Teksti: Simo Koivukoski Kuvat: Manu Pärssinen

**V**ampire 600 on FPGA-pohjainen kiihdytinkortti vuonna 1991 julkaistulle Amiga 600 -kotitietokoneelle. Kortin on tehnyt Bosniaan kotoisin oleva Igor "Majsta" Majstorovic täysin harrastuspohjalta.

Igor Majstorovic kävi digitaalisen elektroniikan ja televiestinnän koulun ja oli hetken harjoittelijana suuressa piirilevytehtaassa. Hänen kotikaupungissaan 12 000 ihmistä työskenteli digitaalisen elektroniikan teollisuudessa, mutta sodan jälkeen kaikki oli mennyttä ja myös Igorin elektroniikkaopinnot lopuivat. Nyt hän on 32-vuotias pienen tyttöävän isä ja työskentelee koulussa maantiedon opettajana.

Igorilla oli pitkään haaveena omistaa Amiga-tietokone. Ensimmäisen Amigan sa hän hankki toukokuussa 2010. Vampire 600 -projekti alkoi tammikuussa 2011. Kiihdytinkortin tekemisessä Igoria motivoi etenkin se, miten tietyt länsimaalaiset kehittäjät suhtautuivat Balkanin alueen harrastajiin. Igor halusi kokeilla, miten vaikeaa kiihdytinkortin tekeminen Amigalle oikeasti olisi.

## Tekniikkaa ja lakitekniikkaa

Koska Igorilla ei ollut aiempaa Amiga-kokemusta, Vampire 600:n suunnittelu vaati runsasta taustatietojen keräämistä. Aluksi tietojen saanti oli hankalaa, koska Amiga-yhteisössä osajien asenne ei ollut järin avulias. Igorille naureskeltiin tai sanottiin, että hän haluaa toisten tekevän kaiken työn puolestaan. Joidenkin mielestä taas koodi puhuu puolestaan eikä

sen toimintaperiaatetta tarvitse selittää.

Onneksi aivan kaikki vanhat osajat eivät ole kynnisiä jääriä, jotka ovat jumiutuneet keskustelufoorumeille. Igorin ajatukset herättivät myös kiinnostusta, ja hän sai paljon korvaamatonta apua projektin loppuun viemiseksi.

Teknisten vaikeuksien lisäksi Igorin kotimaa Serbia ja etenkin sen tulliviranomaiset toivat hankkeelle oman lisävärinsä. Senkin jälkeen, kun Vampire 600 oli palkittu Serbian tasavallan tiede- ja teknologiakilpailussa, oli tulli sitä mieltä, ettei yksi henkilö voi tilata kerralla 50:tä samanlaista piirilevyä. Niitä oltiin jo palauttamassa takaisin valmistajalle Kiinaan, kun Igor sai moninaisten lakiviittausten kautta selvitettyä oikeutensa piirilevyihin. Hän suhtautuu edelleenkin hieman arastellen komponenttien tilaamiseen.

Valmiin kortin ennakkotilaukset käynnistettiin kuitenkin lopulta elokuussa 2013, ja ensimmäiset saivat korttinsa syyskuun aikana. Kuten niin monesti ennenkin Amiga-harrastuksessa, olivat suomalaiset nytkin tilaajien eturintamassa.

## Ohjelmoitavaa tehoa

Aiemmin Amiga 600:n kiihdytinkorttien ideana on ollut korvata alkuperäinen 7 MHz:n kelloaajuudella toimiva Motorola MC68000-suoritin jollakin nopeammalla saman suoritinperheen jäsenenä. Muutamille Amiga-malleille on myös saatavilla PowerPC-suorittimiin pohjautuvia kiihdytinkortteja, mutta niissäkin käytetään ohjelmiston yhteensopivuuden

vuoksi edelleen rinnalla MC680x0-prosessoria.

Vampire 600 on periaatteeltaan täysin erilainen, sillä se ei sisällä lainkaan normaalia keskussuoritinta. Perinteinen suoritin on korvattu digitaalisella mikropiirillä, jonka toiminnan voi ohjelmoida laitteistokuvauskielellä.

Vampire 600:n FPGA-logiikkapiiriksi (field programmable gate array) on valittu Altera Cyclone II, ja sen toiminta on ohjelmoitu emuloimaan MC68000-suoritinta. Emulaatio pohjautuu TG68.C-ytimeksi kutsuttuun laitteistokuvaukseen, jonka Tobias Guebner on alun perin kirjoittanut Minimig (Mini Amiga) -järjestelmää varten. Alastair M. Robinsonin avustuksella mallintamiseen käytetty VHDL-koodi muokattiin Vampire 600:n rautaan sopivaksi. Koodia voi koska tahansa päivittää uudempaan kytkemällä Vampire 600 -kortti pc-tietokoneeseen USB-Blasterin avulla ja käyttämällä Quartus II -ohjelmistoa.

FPGA-logiikkapiiristä johtaa 16-bit-tinen väylä Amigaan, ja signaalitasojen sovittamisen hoitaa lähetin-vastaanottimena toimiva 74ALVT162245-mikropiiri. SDRAM-muistipiiri on kytketty FPGA-logiikkapiiriin 16-bittisellä väylällä, jossa liikennöidään neljän 16-bittisen sanan pituisina sarjoina kerrallaan.

Kiihdytinkortin suunnittelussa on vaurauduttu myös laajennuksiin. Kortissa on neljä erilaista laajennusporttia. Valmiina ovat pin-liittimet MicroSD-kortille ja PS2-hiirelle sekä kaksi lisälaitteporttia, jotka tarjoavat jännite- ja I/O-kytkennät

FPGA:han. Esimerkiksi SL811HS-piiriin perustuvan USB-ohjaimen valmistamista on harkittu.

### Vanhus valonnopeuteen

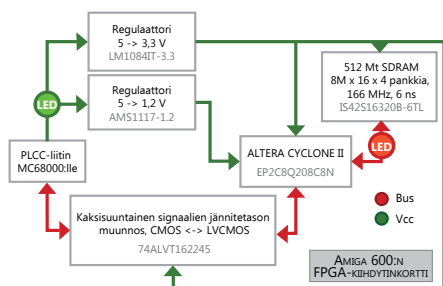
Vampire 600 -kiihdytinkortti asennetaan painamalla sen PLCC-kanta Amiga 600:n emolevyllä olevan MC68000:n päälle, jolloin alkuperäinen prosessori kytkeytyy kokonaan pois käytöstä. Kiihdytinkortti mahtuu täysin Amiga 600:n koteloon sisään, eikä sen olemassaolo näy päällepäin. Erot tulevat esiin vasta sitten, kun koneeseen kytkee virran.

Laajentamaton, yhden megatavun chip-muistilla varustettu Amiga 600 antaa SysInfo-nopeustestissä tulokseksi noin 0,55 MIPSiä (miljoonaa käskyä sekunnissa). Jos koneessa on pelkkää chip-muistia, se jaetaan prosessorin ja erikoispiirien kesken, mikä hidastaa koneen toimintaa. FPGA:n versio 0.1 toimii 87,5 MHz:n kellotaajuudella ja antaa Vampire 600 -kortin 512 megatavun SDRAMista käyttöön viisi megatavua fast-muistia. Fast-muistia käyttää vain prosessori, minkä vuoksi se jo itsessäänkin nopeuttaa konetta. Amigan nopeus kasvaa noin 2,75 MIPSiin.

Kun kiihdytinkortti on käytössä, ohjelmat toimivat nopeammin, ja Amiga 600:lla voi ajaa raskaampia, isommille Amigoille tarkoitettuja sovelluksia. Peli-toiminnan varmistamiseksi osa muistista emuloi pseudo fast -muistia, joka tuli tutuksi jo Amiga 500:n lisämuisteista.

### Ja vieläkin nopeammin

Igor Majstorovicin kotisivulta on ladattavissa Vampire 600:n ytimen versio 1.0. Sen avulla FPGA-logiikkapiiriin kellotaajuus nostetaan 100 MHz:iin ja se ohjelmoidaan matkimaan MC68020-suorittinta, jonka monet Amigan hyötyohjelmat vaativat. Lisäksi emulaatio sisältää välimuistin ja 64 megatavua RAM-muistia. Tämän avulla Amigan nopeus nousee SysInfo-testiohjelman mukaan 6,25 MIPSiin, jolloin nopeusero alkuperäiseen Amigaan on jo melkoinen. Toistaiseksi tämä ydin on kuitenkin epävakaa, mikä todennäköisesti johtuu signaalien synk-



Vampire 600:n piirikaavio.



ronoinnin ja välimuistiemulaation ongelmista.

FPGA:n kellotaajuus ei sinällään juuri vaikuta Amigan nopeuteen, koska TG68.C-koodin suoritus on odotuspohjaista ja riippuu Amigan ja FPGA:n välisestä signaaliikenteestä.

Gunnar von Boehn on tehnyt Nat Ami (Native Amiga) -projektiä varten Apolloytimen, joka on yhteensopiva MC68000-suorittimen kanssa. Tämä ydin toimii eritavalla kuin Vampire 600:ssa käytössä oleva Minimigin TG68.C. Apollo pystyy suorittamaan tehtäviä rinnakkain nykykaisten prosessorien tapaan.

Koska Apollo-ydin ei mahdu sellaisenaan Vampire 600:n FPGA-piiriin pienen muistiin, Gunnar ryhtyi tekemään siitä Vampirelle sopivaa versiota, jonka työnimenä on Phoenix. Hänen päivätyönsä vuoksi Phoenixin kehitys on ollut odotettua hitaampaa ja viime ajat työ on ollut käytännössä pysähdyksissä. Phoenix on noin 70-prosenttisesti valmis.

Phoenix-ytimen avulla Vampire 600:n FPGA pystyisi siirtämään tietoa rinnakkain sekä SDRAMiin että Amiga-väylään. Sen arvellaan pystyvän jopa 100 MIPSin nopeuteen. Se vastaa suunnilleen nopeinta perinteistä Motorola MC68060 -prosessoria, joka on ylikellotettu 80 MHz:n taajuuteen.

### Logiikkaliimaa

Suurin ongelma Amigan FPGA-turbon kehittämisessä on ratkaista kaksisuuntaisten signaalien jännitetasojen sovittaminen Amigan CMOS:n ja Altera Cyclone II FPGA-logiikkapiiriin LVCMOS:n välillä. Laitteistokuvauksen ohjelmoinnissa haasteita tuottavat edelleen signaalien ajoitusten sovittaminen, jännitetasojen säädöstä johtuva viive sekä muistinhalinta. Tämä koskee myös kehitteillä olevaa Phoenix-ydintä.

FPGA:ta ohjelmoitaessa on huomioitava erityisesti se, että ohjelmakoodi ei käänny vain yksinkertaisiksi logiikka-

kytkennöiksi, vaan myös niiden fyysiset sijainnit ja kytkentöjen reititys on huomioitava. Kun piiri tulee niin sanotusti täyteen, tämä voi aiheuttaa ongelmia. Tällöin kääntäjä ei saa enää sijoitettua kytkentöjä optimaalisesti, vaan ne saattavat sijaita hankalasti eri puolilla piiriä. Erityisesti muistisignaalien, välimuistin ja CPU-emulaation sijainnilla on merkitystä ajoitusten kannalta, ja mahdolliset ongelmat vain kertautuvat kellotaajuuden noustessa.

Vampire 600 on mielenkiintoinen uusi askel kiihdytinkorttien rintamalla, vaikka se tällä hetkellä vaatiikin käyttäjältään kärsivällisyyttä ja virittelyintoa. Se on ensimmäinen Amiga-turbo, joka täysin hylkää perinteisen ajatuksen, että kortilla on oltava Motorolan MC68000-perheen suoritin. Pelkästään sekin on hyvä saavutus, että kortin ansiosta koneessa toimii melkoinen määrä ohjelmia alkuperäistä nopeammin. On odotettavissa, että tilanne vielä paranee.

Vampire 600 on avoimen lähdekoodin projekti. Sen piirikaaviot ja lähdekoodi ovat ladattavissa osoitteesta <http://www.majsta.com>. Korttia on myös mahdollista ostaa valmiina: hinnaksi muodostuu kirjoitushetkellä noin sata euroa. Igor valmisti ensimmäiset 50 korttia itse, mutta seuraavat 300 kappaletta kokoaan kanadalainen harrastaja Brian "Kipper2K" Robotham. 🐉

### Miksi juuri Vampire?

Kiihdytinkortin nimi Vampire tulee siitä, että se on Igorin mukaan ainut serbiankielinen sana, joka on maailmalla käytössä. Pääkallologo on taas vanha aiheeseen liittyvä kuva, joka sattui löytymään hänen kiintolevyltään. Kuva on ollut mukana aivan alkuhetkestä saakka. Overclock ready -teksti on toiselle laitteistosuunnittelijalle osoitettu viesti, mutta sillä on todellinenkin merkitys, joka selviää tulevaisuudessa.



## Ei näin! Kotitietokoneesta konsoliksi

*Kotitietokoneet pysäyttivät pelikonsolien ensimmäisen aallon 80-luvun alkuvuosina. Pian konsolit kuitenkin palasivat ja kotimikrovalmistajat halusivat osan niiden markkinoista. Mieluiten mahdollisimman halvalla.*

Teksti: Mikko Heinonen Kuvat: Wikimedia Commons -käyttäjät Bilby ja Evan-Amos.

Vuoden 1983 videopeliromahdus ei syrjäisessä Suomessa juuri aaltoja nostattanut, mutta rapakon takana se oli totisinta totta. Muutamassa vuodessa kehittyneet pelikonsolimarkkinat tulivat täyteen ja ostava yleisö kyllästyi etenkin Atarin laadunvalvonnan puutteisiin. Päivän sana olivat kotitietokoneet. Niillä saisi tehtyä paljon muutakin kuin vain pelattua! Tosiasiassa valtaosalle ne olivat kuitenkin edelleen pelikoneita siinä missä konsolitkin.

Melko pian alkoikin kuulua kummia. Konsolit tekivät paluuta japanilaisin voimin, ja ne kävivät kaupaksi paremmin kuin koskaan. Pohjois-Amerikka antautui Nintendon harmaalle matolaatikolle vuonna 1985 ja muut länsimaat seurasivat pian perässä. Segankin konsoli kävi kaupaksi. Kaikki eivät halunneetkaan tallettaa kakkureseptejä ja olla yhteydessä muihin hakkereihin, vaan pelkkä vaivaton pelaaminen kiinnosti. Nintendon esittelmän lisensointimallin myötä pelikonsolit olivat hyvä bisnes myös valmistajilleen, sillä ne saivat jokaisesta myydyistä pelistä itselleen mukavan siivun.

Tietokonevalmistajat haistoivat rahan. Koska pelikonsolit olivat kotimikroihin verrattuna teknisesti yksinkertaisia, sellainen olisi helppo kääristä kokoon olemassa olevasta laitteistosta. Kolme kuuluisaa kotitietokoneiden valmistajaa väsasikin oman konsolinsa. Yksikään niistä ei aiheuttanut harmaita hiuksia Kauko-idässä.

### Atari XEGS

Ennen suurta romahdusta Atari oli ollut jättiläinen. Pölyn laskeuduttua se kuitenkin jaettiin kahtia ja kotilaitteiden valmistusta jatkoi Tramielin suvun omistama Atari Corporation. Pelien kehitys jäi Atari Gamesiin. Takataskusta löytyi muutamakin vanha konsolimalli, kuten aikanaan mahtavasti menestynyt 2600 ja sen epäonnisemmat seuraajat 5200 ja 7800. Atari Corporation julkaisikin 2600:n ja 7800:n uudestaan vuonna 1986, mutta seuraavana vuonna niiden rinnalle tuotiin myös uusi laite, XE Game System.

XEGS pohjautui Atarin XE-kotitietokoneemalleihin, jotka taas olivat jatkokokehitystä jo vuonna 1979 esitellystä 400/800-mallisarjasta. Suorittimena oli 1,79 MHz:n 6502, äänistä vastasi kolikkopeleistä tuttu POKEY-piiri. Itse asiassa jo Atari 5200 (1982) oli lainailut näitä komponentteja, mutta siitä oli tarkoituksella tehty epäyhteensopiva kotitietokoneiden kanssa.

Atari luotti siihen tosiasiaan, että kotimikroille oli tehty koko joukko moduuleilla toimivia ohjelmia, joita XEGS voisi ajaa ilman muutoksia. Näin uuden konsolin ohjelmistovalikoima olisi olemassa jo sen esittelyhetkellä. XEGS:stä tehtiin kaksi mallia: perusmallissa sai pelkän keskusyksikön ja Atarin legendaarisen huonon peliohjaimen; deluxe-mallin mukana toimitettiin myös näppäimistö ja valopistooli. Näppäimistön liittämällä XEGS muuttui käytännössä Atari 65XE -koti-mikroksi, johon oli mahdollista liit-

tää myös sen lisälaitteita.

Laitteen suunnittelu oli siis kierrätystä, ja se näkyi kaikessa. Iso osa peleistäkin myytiin vanhoissa XL/XE-koteloissa, joihin oli vain liimattu uusi tarra XEGS-yhteensopivuudesta. Tämä puhui omaa kieltään myös siitä, miten tuoreita pelit olivat. Kun Nintendo ja Sega toivat samaan aikaan markkinoille uutuuden toisensa jälkeen, XEGS:llä ei ollut todellista mahdollisuutta kilpailla.

Atari oli aliarvioinut tuoreen ohjelmiston merkityksen koneen menestykselle. Vaikka japanilaisilla ei ollut käytössään juuri tätä kummempaa rautaa, heillä työskentelivät alan parhaat ohjelmistokehittäjät. Sellaisia Atari ei onnistunut houkuttelemaan, vaan XEGS:lle tarjottiin lämmitettyä eilistä huttua. Mikrobitissä Markku Alanen vertasikin Atarin esitystä keisarin uusiin vaatteisiin.

XEGS jäi yhdessä 2600:n ja 7800:n uusintakierrosten kanssa pieneksi sivujuonteeksi pelaamisen historiassa. Laite ei ole mahdollottoman harvinainen: ehjää sellaisesta saa verkkohuutokaupoissa maksaa noin sata euroa.

### Commodore 64 Games System

Maailman suosituinta kotimikroa yritettiin väantää pelikonsoliksi kahteenkin kertaan. Ensimmäinen yritys oli Japanin markkinoille tarkoitettu Commodore MAX Machine vuodelta 1982. Se oli käytännössä yksinkertaistettu, muistiltaan riisuttu ja kalvonäppäimistöllä varustettu C64. Teknisestä etumatkastaan huolimatt-

ta laite kuitenkin jäi paremmin laajennettavan VIC-20:n jalkoihin.

Epäonnistumisen jäljiltä Commodore malttoi miltei kahdeksan vuotta ennen kuin yritti uudelleen. Vuonna 1990 konsolimarkkinoiden houkutus kuitenkin kävi liian suureksi ja vanha sotaratsu päätettiin vielä kerran valjastaa uuteen tehtävään. Syntysanat lausuttiin C64-konsolille, joka sisältäisi vain pelaamisen kannalta välttämättömän. Näppäimistö ja laajennusportit saisivat lähteä, ja pelit tarjottaisiin vain moduuleilla. Näin päästäisiin eroon myös kuusnelosen hitaista massamuisteista.

Lehdistö ja suuri yleisö suhtautuivat alusta asti epäluuloisesti kuusnelosen konsolointiin. Moduulien mahdollistamat suuret ja nopeasti latautuvat pelit nähtiin positiivisena asiana, mutta niiden pelaaminen onnistuisi oikeallakin kuusnelosella. Sellaisella voisi lisäksi käyttää myös levyille ja kasetille tarkoitettua ohjelmistoa. Sitä taas oli saatavana murto-osalla pelimoduulien hinnasta. Commodore puhui suurellisesti ”sadoista työn alla olevista peleistä”, mutta käytännössä niitä julkaistiin alle 30, joista vain noin kolmannes oli uusia eikä uudelleen pakattu- ja tavallisia C64-pelejä.

Eikä tässä vielä kaikki. Jouluna 1990, jolloin C64-konsolin oli määrä ottaa markkinat haltuunsa, samoista pennoista kamppaili myös uusi 16-bittinen Sega Megadrive. Sen pelien parhaimmisto laitto jo Amigankin tiukoille, liki kymmenvuotiaasta kuusnelosesta puhumattakaan. Tarvittiin melko erikoinen yhdistelmä niukkaa budjettia ja ymmärtämättömyyttä, että Games System lähti kaupasta asiakkaan mukaan.

C64GS kävikin kaupaksi suunnilleen ennakoarvioiden mukaisesti, eli ei käytännössä lainkaan. Arviot valmistusmäärästä liikkuvat muutamissa tuhansissa, eikä konetta ilmeisesti myyty kuin jokuisilla markkinoilla - joista yksi oli tietenkin Commodore-valtio Suomi. Onnistuin itse löytämään toimivan Games Systemin kohtuuhintaan viitisentoista vuotta sitten, mutta nykyään sellaisesta saa varautua jo pulittamaan hyvänlaisen kasan euroja.



### Amstrad GX4000

Ei kahta ilman kolmatta. Commodoren kanssa samoihin aikoihin kalassa oli myös Amstrad, joka oli etenkin Isossa-Britanniassa lujittanut asemansa kaikenlaisen viihde-elektronikan markkinoijana. Meillä tunnettujen CPC-kotimikrojen lisäksi Amstradilla oli tarjolla myös televisioita ja äänentoistolaitteita, joten pelikonsoli niveltäisiin tähän pakettiin luontevasti.

Toisin kuin Commodore, ei Amstrad ottanut konsolinsa pohjaksi suoraan vanhaa kotimikroa vaan viritteli sitä ensin hieman. CPC Plussaksi kutsuttu päivitys lisäsi koneeseen sprite-grafiikan ja vierityksen, tuplasi värien määrän 16:sta 32:een ja laajensi väripaletin peräti 4 096 mahdollisuuteen. Äänille saatiin DMA, ja muutenkin kuusivuotiaasta laitteistoa päivitettiin niin paljon kuin se oli kohtuudella mahdollista. Suorittimena säilyi Z80, ja Plus-koneet olivat edelleen yhteensopivia vanhan ohjelmiston kanssa. Tärkeä lisäys oli tietenkin myös portti pelimoduuleille.

Internetin jälkiviisas mielipide on, että GX4000 olisi voinut joissain olosuhteissa menestyäkin. Sitä myytiin alusta asti halvalla - lähtöhinta oli vain 99 punttaa -, eivätkä sen tekniset ominaisuudet olleet mitenkään toivottomat. Valitettavasti vain mitään muita myyntiargumentteja ei ilmaantunut. Pelejä ilmestyi vain vajaat 30, ja nekin saatiin markkinoille hyvin hitaasti. Osasyynä tähän oli Amstradin käyttämä hidas piirilevytehdas. Lisäksi tästäkin pienestä pelimäärästä iso osa oli suoraa kierrätystä CPC:ltä.

Lehdistön mielen-

kiinto taas kohdistui 16-bittisiin kotimikroihin ja pelikonsoleihin, joten Amstrad ei saanut laitteelleen juurikaan näkyvyyttä. Halpa hinta myös näkyi laitteessa. Etenkin sen ohjaimet olivat erittäin muovisen tuntuiset.

GX4000 sai osakseen ehkä aiheetonkin pilkkaa. Olemattoman myynnin vuoksi sen hinta laski nopeasti jopa alle 30 punnan ja Game Zone -lehti jakoi niitä palkinnoksi kuukauden huonoimmasta lukijakirjeestä. Laitteesta tuli vitsi, jolloin Amstrad teki johtopäätöksensä ja veti projektilta töpselin seinästä. GX4000 eli vain syksystä 1990 loppuvuoteen 1991. Nykyisin toimiva sellainen liikkuu samoissa hinnoissa kuin Atari XEGS, mutta harvalukuisista peleistä saa maksaa helposti saman verran kuin itse konsolista.

### Suutarit ja lestit

Kotitietokoneiden valmistus on luonteeltaan hyvin erilaista kuin pelikonsolien tehtailu. Hyvin tehty ja suosittu kotimikro houkuttelee luokseen osaavia kehittäjiä ja ohjelmistoa, jotka taas myyvät lisää koneita. Konsolivalmistajan, etenkin sellaisen jolla ei itsellään ole vahvaa pelikehitysosaamista, on ensin vakuutettava pelitalot tuotteestaan ja sitten tarjottava niille riittävästi tukea. Vain tällä tavoin voidaan luoda niitä paljon puhuttuja ekosysteemeitä, joista syntyy koneita myyviä huippupelejä.

Sekä Atari, Commodore että Amstrad tuntuivat olettavan, että konsolin tekemiseen riittää pelkkä laitteen valmistaminen. Aivan erityisesti niiden yritykset ansaitsivat epäonnistua siksi, että kohdeyleisöä aliarvioitiin kierrättämällä jo parasta ennen -päivänsä ohittanutta rautaa. 🐱





## Retrofuturismi Aika tuleva ei koskaan palaa

*Miksi vuosi 2000 ei ollut kuin vuosi 2000? Tieteiskirjailijat Arthur C. Clarke ja Isaac Asimov visioivat tulevaisuutta aktiivisesti ja osuivat monessa asiassa oikeaan. Vielä useampi toivekuva vuodesta 2000 meni kuitenkin pieleen.*

Teksti: Jukka O. Kauppinen

Kuvat: Risto Mäki-Petäys, Jukka O. Kauppinen, Wikimedia Commons, Retro-Futurismus.de

**T**ulevaisuus tekee ihmisen toiveikkaaksi. 1900-luvun alussa ja vielä 1950-luvulla nähtiin vuosituhanen vaihe aikana, jolloin kaikki haaveet toteutuvat. Ihmiset ovat nähneet sodan järjettömyyden ja elävät rauhassa. Tai jos eivät, taivaalla kiitävät vakoojasilmät havaitsevat uhat ja tekevät sodista mahdolltomia. Köyhyys on teknologian ansiosta hävitetty, ja kaikki elävät yltäkylläisyydessä. Niin autot kuin lentokoneetkin toimivat ydinvoimalla, ja säännöllinen rakettiliikenne vie Kuuhun, Marsiin ja kauemmaskin. Moni ennustus toki toteutui mutta vielä useampi ei.

Missä on ydinautoni, robottipalvelijani tai lentoreppuni? Miksi en ole ollut kesälomalla Kuussa tai viikonlopun pyrähdyksellä kiertoratahotellissa? Miksi niskassani ei ole plugia, jolla kytkeytyä virtuaalitodellisuuteen? Missä ovat kaupunkiliikenteestä huolehtivat ilmalaiivat ja Maa-Jupiter-reittiraketit? Missä on kaikki se, mitä minulle luvattiin? Kuka varasti tulevaisuuteni?

### Retrofuturismi luotti tulevaan

Retrofuturismiksi kutsutaan menneiden vuosikymmenten ja -satojen tapaa nähdä tulevaisuus teknologiaa ja kehitystä ihannoivan linsin läpi. Se on 1950–1970-luvuilla voimakkainta kauttaan elänyt suuntaus, jossa teknologian uskottiin tuovan ratkaisun lähes kaikkiin ongelmiin. Huomispäivän maailman piti olla värikäs onnela, jossa työnteko ja vapaa-aika kukoistavat täydellisessä sopuolosuhteissa.

Retrofuturismin kultakauden arkkitehtuurin ja designin suuntaukset tunnustaa lentävistä muodoista ja rohkeudesta. Tyyliuntaan sopivia viittauksia löytyy myös 1800-luvun ja 1900-luvun alun taiteesta ja tieteestä. Eräs vanhimmista tunnetuista suomalaisista tieteisnovelleista kertoo Helsingistä, jossa ihmiset matkivat lentobusseilla ja omilla pienillä lentolaitteillaan.

Tämä kultainen aikakausi elää nykyään taiteena, tyylinä ja muotoiluna, joissa paitsi ihailaan menneiden aikojen

näkemyksiä, myös tulkitaan niitä uudelleen. Jotkin modernit kuvakulmat ovat kuin tieteiskirjallisuutta: ikkunoita vaihtoehtoisin maailmoihin, joissa vanhat ennustukset ovat tulleet todeksi. Retrofuturismi elää vahvasti myös viktoriaanisen Britannian aikakautta ihannoivassa steampunkissa.

Eräs retrofuturismin suurista taiteilijoista ja visionääreistä oli yhdysvaltalainen Arthur Radebaugh (1906–1974), jonka jälkipolvet ovat nimenneet tyyli-suuntansa da vinciksi. Radebaugh löysi 50-luvulla kuvitustöihinsä huikeita futuristisia teemoja, joita hän esitteli sanomalehtien sarjakuvasisivillä ilmestyvässä Closer Than We Think -sarjassaan. Niissä hän loitsi 19 miljoonalle lukijalleen häikäiseviä näkymiä tulevaisuuden kaupungeista, arkkitehtuurista ja arkipäivästä. Osa näistä on sittemmin tullut todeksi.

Radebaugh olikin eräs kulmakivi aikakautensa luottavaiselle näkemykselle tulevaisuudesta. Silti meillä ei valitettavasti vielääkään ole loma-asuntoja Kuussa,

rakettirepulla lentäviä postinkantajia tai napin painalluksella toimivia oppimiskojeita kouluissa.

Osa retrofuturismin viehätystä on paluu jo kertaalleen unohtuneiden, parempaa maailmaa maalaavien uskomusten pariin ja ylihuomisen maailman peilaaminen niiden kautta. Mitä aiemmin kuviteltiin, mitä saatiin ja minne voimme päästä?

### Pilvikaupunkeja ja taivastorneja

Taivaalla leijuvia ja meren pohjassa sijaitsevia asuntoja kuvitelleet menneisyyden visionäärit pitäisivät nykypäivän kaupunkeja tylsinä ja takapajuisina. Muutamaa satunnaista yritystä lukuun ottamatta arkkitehtuuri on näet kulkenut vuosikymmenten ajan väärään suuntaan. Retrofuturistisen arkkitehtuurin vallitsevia piirteitä olivat muotojen pyöreys ja lennokkuus sekä massiiviset mittasuhteet. Tämän päivän pilvenpiirtäjät olisivat taitavia kurotteluvien, linjakkaiden ja kutsuvien kolossien juurella pelkkiä mummonmökkejä.

Sen sijaan että kotimme olisivat pylväiden nokassa nököttäviä lentävien lautasten muotoisia upeita taloja tai Blade Runnerin mahtavia, neonhohtoisia pilvenpiirtäjiä, meitä kirottiin 70-luvun betonilähiöillä. Elokuvan slummit sopii unohtaa, mutta ikkunan eteen parkkeeraava lentävä kiinalainen gondoliravintola kelpaisi. Tai korttelit, joita voidaan siirtää rataakiskoja pitkin aina, kun asukkaat kyllästyvät ikkunanäkymäänsä.

Onneksi visiot muuttuvat hetkittäin todeksi – joskin vain paikoittain ja liki vahingossa. General Motorsin tehtaalla Detroitissa oli upea kolmella jalalla seisova ufovesitorni, joka sopisi minkä tahansa B-luokan scifipätkän kulissiksi. Los Angelesin kansainväliselle lentokentälle rakennettiin vuonna 1961 ravintolarakennus, joka muistuttaa lentokentälle laskeutunutta ufoa.

Toiveikasta näkemystä edusti myös vuoden 1964 New Yorkin maailmannäyttely, joka lienee useimmille tuttu Men in Black -elokuvasta. Maailmannäyttelyn teema oli ”rauha ymmärryksen kautta”, ja monen maan osastot kuvasivatkin näkemystä onnellisemmasta, teknologisemmasta tulevaisuudesta. Etenkin näyttelyn upea maapallo sekä ufopaviljongit ja näkötornit ovat jääneet länsimaisen arkkitehtuurin historiaan. Vain pari vuotta aiemmin Seattleen valmistunut Space Needle -torni kertoo upeaa kieltään tähtiin kurottavan arkkitehtuurin historiaa.

Tulevaisuuden kaupungit ja asumis-



Los Angelesin lentokentän Theme Building on virallisesti kulttuurihistoriallinen monumentti. Kuva: Wikimedia Commons / Monkeytime\_Brachiator.



New Yorkin maailmannäyttelyn kuuluisat maamerkit. Kuva: Wikipedia Commons / Fredverillo.

muodot eivät kuitenkaan toteutuneet. Vuoden 2000 sijaan saimme Hervannan.

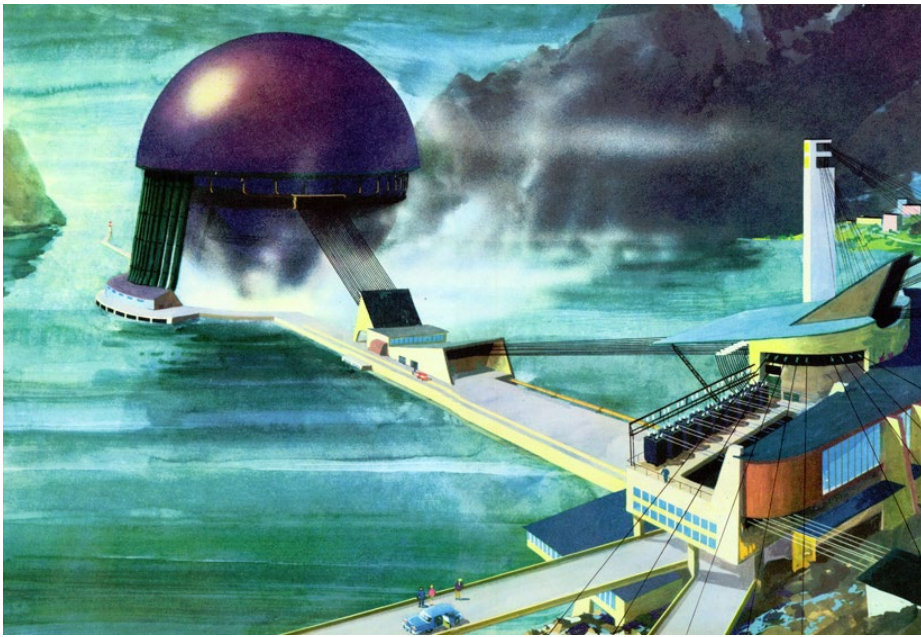
Postikortti vuodelta 1910 kuvaa tulevaisuuden New Yorkia. Näkemys on lähellä arkkitehti Harvey W. Corbettin vuonna 1925 rakentamaa mallia tulevaisuuden kaupungista, joka paitsi kurottaa taivaalle, myös kaivautuu syvemmälle. Talojen katoilla oli pystysuoraan nousevien lentokoneiden ja zeppelinien kenttiä. Rakennukset olivat korttelin kokoisia kokonaisuuksia, joiden katutasoilla sijaitsivat liikkeet ja ylhäällä asunnot. Kadut olivat leveitä, kauniita esplanadeja, ja autoliikenne kiisi maanlaisilla väylillä useassa kerroksessa, sähköjunat ja huoltoväylät alimpina. Moottoritiet oli koristeltu niiden ylle rakennetuilla puistoilla.

Suomalaisen retrofuturismin näyttävien saavutukset ovat arkkitehti Matti Suurosen 60-luvulla suunnittelemat Futuro-asunnot. Avaruustaloiksi kutsutut





Suomalaista arkkitehtuuria 60-luvulta: Futuro-talo.



Turvallista lähiatomienergiaa kivenheiton päästä.  
Kuva: Retro-Futurismus.de / Eberhard Binder-Staßfurt.

Futurot olivat muovisia, helposti pystytettäviä loma-asuntoja, joiden kokeellinen ja mieleenpainuva design sai laajalti huomiota maailmalla. Valitettavasti 1970-luvun öljykriisi tuhosi suomalaisen ufotalon maailmanvalloitus suunnitelmat. Viimeinen Suomessa säilynyt Futuro oli pitkään kesämökkinä Puulavedellä, missä se ehti lähes tuhoutua säiden armoilla. Vuonna 2012 se kuitenkin pelastettiin Espooseen näyttelykeskus Weegeehen.

### Atomikoti ja robottipalvelijat

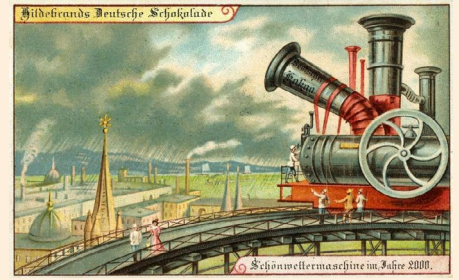
Futuristisen arkkitehtuurin ohessa historia vei myös tulevaisuuden kodin. Enkä puhu pelkästään merenpohjakaupungeista tai antigravitaatiomökeistä vaan niinkin yksinkertaisesta asiasta kuin ruohonleikkuusta. Jos asuisin retrofuturistisessa omakotitalossa, atomiruohonleikkurini pitäisi pihan tiptop-kunnossa, eikä sitä tarvitsisi ladata koskaan.

Jotain toki saatiin. Jääkaappi ja pyykkipöytä mullistivat 50-luvulla perheen-

äitien elämän. Samaan aikaan kun niitä kannettiin amerikkalaiskoteihin, suunnitteli IBM sisäkkörobotteja tekemään yksinkertaisia kotitöitä. Nykyiset imuri ja ruohonleikkurirobotit ovat kuitenkin kurja korvike palvelijaroboteille, jotka huolehtisivat siisteydestä, huvittaisivat isäntäväkeä ja opettaisivat lapsia.

Internet-yhteydellä varustetusta jääkaapista on niin ikään pitkä matka rehtiin älykeittiöön, jossa palvelusrobotti käskisi keittiötä muhentamaan herrasväelle atomeista aamiaista muutamassa sekunnissa. Kulinaristit voivat toisaalta olla tyytyväisiä siihen, etteivät ravintopillerit tai ruokarobottilähetin toimittamat, itsestään lämpiävät valmisateriat ole vielä korvanneet ruuanlaittoa ja herkuttelua.

Atomivoima on halpaa, turvallista ja helppokäyttöistä. Autot ja lentokoneet toimivat pikkuruuisilla atomireaktoreilla, eikä uusiutuville energianlähteille ole tarvetta, kun kotoisaa lähisähköä saadaan kotitalon eteen ankkuroiduista voi-



### Sääkone on yhä keksimättä

Tieteiskirjallisuuden ja futurologien pitkäaikainen haave, säänhallinta, odottaa yhä toteutumistaan. Ilmastoa ei pystytä muokkaamaan, vaikka paikallissäähän voidaan yrittää vaikuttaa. Tämä ei kuitenkaan onnistu edes Haarpilla vaan pilviä kylvämällä. Näin voidaan joko yrittää luoda tai estää sadetta, kuten tehtiin onnistuneesti muun muassa Moskovan ja Pekingin olympialaisissa.

Ilmastonmuokkauksineiden puute ei kuitenkaan johdu yrityksen puutteesta. Mitä enemmän ilmastosta tiedetään, sitä vähemmän sitä ymmärretään. Tutkimustyö on vaativaa ja säätilan muokkaukset voivat aiheuttaa ketjureaktioita, joilla saattaa olla arvaamattomia vaikutuksia. Saamme tuskin varsinaisia sääkoneita vielä pitkään aikaan – etenkin, kun ilmastonmuutoksen tutkiminen vaatii panostusta planeetanlaajuisessa mittakaavassa.

On siis vielä matkaa siihen, että eläisimme täydellisesti sääillistetyssä kaupungissa kuten Mega-City Onessa. Yli satavuotiset haaveet kupujen tai valtavien lasikattojen alla sijaitsevista kaupungeista, joissa keli on aina täydellinen, ovat yhä vain kaukaisia haaveita.

maloista.

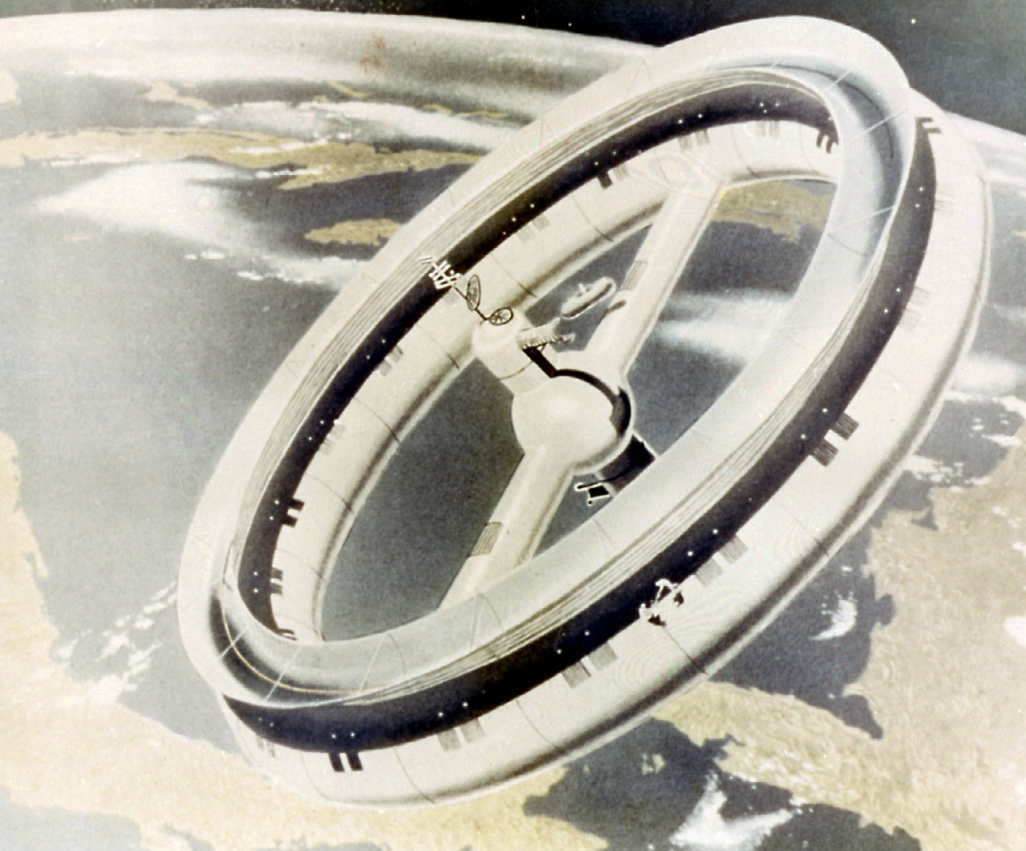
Kun sähköä tuotetaan, sitä tuotetaan urakalla. Vaikka ydinvoima onkin täysin turvallista, voidaan sähkön saatavuus varmistaa kiertoradalle sijoitetuilla aurinkopaneelilla, joiden sähkö siirretään maanpinnalle mikroaaltoina.

Philco-Fordin 1970-luvulla valmistama televisio on puhdasta retrofuturismia niin muotoilultaan kuin väritykseltäänkin.

Osa vanhemmistakin ennustuksista osui maaliinsa. Jo 1800-luvulla brittiläiset herrasmiehet haaveilivat siitä, kuinka aamuteetä voisi siemailla pedissä katsellen kuvagramfonin seinälle heijastamaa uutislähetystä. Kuinka todeksi tuo kävikään! Olemme myös saaneet sähköiset kirjastot ja langattomat puhelimit, joten aivan kaikki ei mennyt pieleen.

Professori Eero Aarnion Pallotuoli (1963) on yksi suomalaisen designin tunnetuimpia ja maailmalla menestyneimpiä tuotteita. Se kuvasi jokseenkin täydellisesti retrofuturistisen muotoilun ajatus-





Wernher von Braunin suunnittelema avaruusasema vuosimallia 1952. Kuva: NASA.

maailmaa. Pyöreä, lasikuituinen istuin jälkeläisineen on kelvannut jopa Playboy-kanteen sekä James Bond- ja Men in Black -elokuviin.

Jo 1900-luvun alussa haaveiltiin matka- ja kuvapuhelimista. Telefooniin vain kiinnitetään käsipeilimäinen näyttö, iskeään luurit korville ja juttu luistaa. Musiikkisoittimet latauspalveluineen ovat jo ylittäneet entisajan visionäärien haavekuvat. Mutta miten olisi käynyt heidän tulevaisuudenuskonsa, kun kuulokkeista olisikin soinnut henkevä sinfonian sijaan Kraftwerkia, Sigue Sigue Sputnikia tai Alien Sex Fiendä?

Muoti ja arkkitehtniikka ovat saaneet menneen lupaukset osittain kiinni, ehkä menneet ohikin. Foliopuvut, hopeiset kookasut pilottilaseineen tai neonväreissä loistavat, ihoon liimautuvat arkikerrastot eivät ole yleistyneet. Sen enempää kemia kuin korsettitkaan eivät ole vielä teh-

neet meistä fyysisesti täydellisiä eivätkä identtisiä.

Menneisyyden retrofuturistit olisivat varmasti kummissaan tämän päivän konemusiiikki- ja fetissibileissä sekä skifistisen musiikin että vaateteollisen kehityksen takia. PVC- ja kumiasut, korkosaappaat ja räiskyvät moonbootsit ovat jo arkipäiväisiä asusteita. Tämän päivän tulevaisuusfriikki tosin pukeutuu kimaltelevan hopean sijaan enemmänkin avaruuskumiin ja niukkuuteen.

### Lomani kiertoratahotellissa

Jos maailmassa olisi lainkaan oikeutta, olisivat scifi-kirjailijoiden visioimat kiertoratahotellit jo totta. Laittaisın kolikkoni pinoon ja lentäisın rakettikoneella lähimmälle avaruushissille, joka nostaisi minut



Kuva: Wikimedia Commons / Saillko.

kiertoratahotelliin viettämään pitkää viikonloppua. Sieltä kävisin päiväretkellä Kuussa kummastelemissa ensimmäisen kuulennon laskeutumispaikkaa ja ostaisin kuuhattaran Apollo 11 -huvipuistosta.

Mutta ei! Ei siitäkään huolimatta, että jo toisella vuosisadalla kreikkalainen kirjailija Lucian loi tarinan laivasta, jonka miehistö huuhtoutui myrskyssä Kuuhun. Kului toista tuhatta vuotta, ennen kuin Jules Verne kirjoitti tunnetuimmat teoksensa Maasta Kuuhun (1865) ja Kuun ympäri (1870), jotka innoittivat lukemattomia ihmisiä katsomaan kaipaavasti yötaivaalle.

Ensimmäiset varteenotettavat suunnitelmat avaruusasemista teki saksalainen rakettitiedemies Wernher von Braun, joka suunnitteli avaruuslentoja 1930-luvulla. Von Braun lupasi eräälle rakettihävittäjän koelentäjälleen: "Sinä ohjaat ensimmäistä alustani Kuuhun!" Alusta lensi kuitenkin amerikkalainen astronautti vuonna 1968. Tässä välissä von Braun rakensi myös maailman ensimmäisen ballistisen risteilyohjuksen.

Varsinaista avaruusturismia ja säännöllistä planeettainvälistä liikennettä käsittelevästä vasta Robert A. Heinleinin vuoden 1957 novellissaan The Menace from Earth. Osittain Heinleinin ansiosta avaruusturismin uskottiinkin olevan arkipäivää vuoteen 2000 mennessä.

Usko avaruuslomiin oli niin vahva, että maailman suurimmalla lentoyhtiöllä Pan Amilla oli oma jonotuslista ensimmäisille lomalennoille Kuuhun. Pan Am arvioi, että kuulentojen hinta saattaisi olla tähtitieteellinen, mutta tästä huolimatta yli 93 000 ihmistä ilmoittautui First Moon Flights -klubin jäseniksi. Klubi purkautui vasta Pan Amin konkurssissa vuonna 1991.

Kun Yhdysvallat voitti kilpajuoksun Kuuhun, romahti sotilaallinen ja kaupallinen kiinnostus avaruuslentoihin ja avaruuslomiin. Avaruussukkuloiden valmistaja Rockwellin suunnitelma sukuloiden ruumaan asennettavasta 72 ihmisten matkustamosta hautautui arkistoihin. Ensimmäinen siviili pääsi avaruuteen vasta 1984, kun lentokonetehdas Mc-



Kuva: Retro-Futurismus.de.



Kuva: Wikimedia Commons / Housing Works Thrift Shops.



Kuva: Retro-Futurismus.de / Klaus Bürgele.

Donnell Douglas maksoi 40 000 dollaria Charles D. Walkerin sukkulalennosta. Varsinainen avaruusturismi alkoi vuonna 2001, kun Dennis Tito lensi venäläisellä Sojuz-avaruusaluksella kansainväliselle avaruusasemalle.

Kirjoitushetkellä avaruudessa on käynyt vasta seitsemän oman lentonsa maksanutta turistia, joista yksi on pelisuunnittelija Richard "Lord British" Garriott. Tällä hetkellä Sojuzit eivät lennä turisteja, mutta Virgin Galactic aikoo aloittaa turistilennot Scaled Compositesin aluksilla lähivuosina. Virgin haaveilee myös avaruushotelleista ja kuulennoista.

Kaupallisia avaruuslentoja ja kiertoratahotelleja juonivat muutkin toimijat, joista yhden Genesis I- ja Genesis II -prototyyppihotellit ovat edelleen Maan kiertoradalla. Jos hyvin käy, näemme lisää avaruusturisteja jo lähivuosina. Silti: liian vähän, vuosikymmeniä liian myöhään.

Olisin jo halunnut ehtiä lomailemaan taivaalle haukkamaisella avaruusaluksella, jollaisten piti lentää Marsiin jo 50 vuotta sitten.

### Avaruushissi on portti avaruuteen

Ajatuksen avaruushissistä keksi venäläinen tiedemies Konstantin Tsiolkovski. Hän ehdotti jo vuonna 1895 hissejä, joiden avulla olisi mahdollista nousta maan pinnalta avaruudessa sijaitseville geostationaarisella kiertoradalla oleville asemille. Ajatusta kehitti eteenpäin tieteiskirjailija Arthur C. Clarke teoksessaan Paratiisin suihkulähteet (1979) ja vuonna 1981 julkaisemassaan tutkimuksessa.

Avaruushissit eivät ole pelkkää scifiä. Useat tutkijaryhmät, Nasa mukaan luki-

en, kehittävät jatkuvasti materiaaleja ja teknologioita, joiden avulla avaruushissi voitaisiin rakentaa ja ankkuroida. Hissit kiinnostavat ennen kaikkea käyttökustannusten edullisuuden takia. Rakentaminen maksaa maltaita, mutta hissien ansiosta lastien rahtaaminen kiertoradalle olisi edullista. Tämä mahdollistaisi avaruusmatkailun uuden aikakauden, ehkä jopa pysyvät asunnot ja mahdollisuuden kaipaamiini viikonloppulomiin Kuussa!

### Liikenne vuonna 2000

Kesäloma vuonna 2000 alkaa, kun robotitaksi noutaa minut pilviin kohoavan asuntoni ovelta. Taksi vie minut magneettiraiteita pitkin läheiselle liikenneasemalle. Siellä on monikerroksisia maanalaisia metrotunneleita, junia, levitaatiobussien ja robotitaksien pysäkkejä ja ylimpänä lentoasema. Halutessani voisin kiittää ylääänimatkustajakoneella tai pikavuororaketilla tunnissa Maapallon toiselle puolelle, mutta tänään suuntaan Kuuhun, Finnairin suoralla Helsinki-Luna City 1 -raketilla.

Tulevaisuuden liikennevälineet ovat jääneet monella tavalla haavekuvien asteelle. Vielä 50-luvulla etenkin yksiraiteiset eli monorailit nähtiin tulevaisuuden kaupunkiliikenteen välineenä. Perinteinen auto veti kuitenkin pisimmän korren. Yksiraiteisten surullista kohtaloa pelataan Simpsonien jaksossa "Joko säteilee" (Marge vs. the Monorail), jossa liukkaat myyntimiehet kauppaavat ylihintaisia ja sekundaalatuksia yksiraiteisia varakkaille kaupungille.

Yksiraiteisissa on lumoa. Miten vil-

### Arthur C. Clarken arvioita tulevaisuudesta

Arthur C. Clarke (1917–2008) on yksi tieteiskirjallisuuden arvostetuimmista visionääreistä. Clarke visioi teoksissaan ihmiskunnan teknistä edistymistä ja kirjoitti runsaasti tieteellisiä artikkeleita, joissa on paljon oikeaan osuneita visioita.

- In Transit to Earth -novelli (1971) ennusti ensimmäinen Mars-lennon vuodelle 1994.
- Prelude to Space -kirja (1951) ajoitti ensimmäisen ihmisen Kuussa vuodelle 1978.
- Novelli The Ghost from the Grand Banks (1990) ennusti vuoden 2000 digitaalisen millenniumkaoksen.
- 1950-luvulla Clarke ennusti maailmanlaajuisen yhteisen kirjaston olevan totta vuonna 2005. Www, digitaaliset kirjat ja Googlen digikirjasto ovat varsin hyvä askel siihen suuntaan.
- Vuonna 1974 Clarke mietti tulevaisuutta Australian yleisradion haastattelussa. Hän arveli, että vuonna 2001 muun muassa verkkopankit ja nettishoppailu olisivat arkipäivää. "Silloin ihmisillä on käytössään kotitietokoneita ja pääte, jonka kautta voi keskustella, varata teatteriliput ja hakea elämässä tarvitsemaansa tietoa. Kaikki tämä on aivan yhtä luonnollista kuin puhelin meille, ja tietoverkkojen ansiosta voimme elää missä haluamme. Kuka tahansa voi tehdä työtään lähes missä tahansa maapallolla tietokoneen ja tietoverkkojen avulla. Jonain päivänä aivokirurgi voi operoida Edinburghista käsin Uudessa-Seelannissa sijaitsevaa potilasta."

liltä kaupunkiliikenne näyttäisikään, jos gyroskoopin vakauttamattomat tai magneettiraiteiset yksiraiteiset olisivat korvanneet raitiovaunut. Muita villien visioiden joukkoliikennemuotoja olivat tyhjiötunneleissa kiitävät junat, valtavat ilmatyynyalukset ja leijuvat valtameriristeilijät, puhumattakaan mereltä suoraan rautatiekiskoille nousevista rahtilaivoista.

Näiden visioiden toteutuminen näyttää kuitenkin yhtä todennäköiseltä kuin moottoriteiden muuttuminen tarpeettomiksi tai levitaatiojalkakäytävien ja henkilökohtaisten lentoreppujen yleistyminen. Viimeksi mainituista on haaveiltu populaarikulttuurissa ainakin 1920-luvulta saakka, niin supersankarien kuin postinkantajienkin työkaluna ja liikennevälineenä.

Lentoliikenteen tulevaisuutta visioivat sekä Jules Verne 1800-luvulla että insinöörit myöhemmin aikoina. 1930-luvun lopussa saksalaistutkija Eugen Sänger aloitti rakettimatkustajakoneiden suunnittelun, mutta Saksan ilmailuministeriö värväsi miehen suunnittelemaan rakettipommikoneita. Sänger muistetaan erityi-

### Isaac Asimovin ennustuksia

Tieteiskirjailija, professori Isaac Asimov mietti vuoden 1964 maailmannäyttelyssä, millaiselta maailma näyttää vuonna 2014. Asimovin arviot osuivat hämmästyttävän oikeaan.

- Ihmiset vetäytyvät yhä kauemmas luonnosta. Valot syttyvät pehmeästi ja eri väreissä ja muuttuvat napin painalluksella.
- Robotit eivät ole vielä kovin yleisiä, mutta niitä on olemassa. Tietokone osaa esimerkiksi kääntää venäjän kieltä englanniksi. Puutarhanhoitorobotteja on kehitteillä.
- Elokuvia esitetään kolmiulotteisesti.
- Aurinkovoimaa käytetään sähköntuotantoon.
- Puhelinviestinnässä on kuva ja ääni. Sähköisiä näyttöjä voidaan käyttää kirjojen lukemiseen ja valokuvien katseluun. Satelliittien ansiosta voi soittaa jokaiseen maailman kolkkaan.
- Seinänäytöt ovat korvanneet tavalliset televisiot.
- Miehitämättömät alukset ovat käyneet Marsissa, mutta miehitetyn retkikunnan lähettämiseksi tehdään töitä.
- Ihmiskunta kärsii ikävystymisestä, ja sillä on vakavia psyykkisiä, emotionaalisia ja sosiaalisia seurauksia.
- Laitteissa ei ole sähköjohtoja, sillä niiden käyttövoima tulee pitkäikäisistä akuista.
- Puhelimella voi lukea kirjoja.

sesti historian ensimmäisen avaruuslentokoneen, Silbervogelin, suunnittelijana. Silbervogel ei koskaan lentänyt, mutta Sängner jatkoi tutkimustyötään liittoutuneiden leivissä. Koneen perintö eli myöhemmin Nasan avaruussukkulassa.

Kalifornian teknologiainstituutin professori Qian Xuesen jatkoi Sängnerin linjoilla vuoden 1949 raketimatrustajakoneen suunnitelmillaan. Qianin kone olisi lennättänyt matkustajat Yhdysvaltojen halki 45 minuutissa. Yliäänimatrustajakoneet Concorde ja Tupolev Tu-144 jäivät ilmailuhistorian kuriositeeteiksi, kun ilmaliikenne kehittyi Boeingin 747:n ja Airbusin A380:n suuntaan. Taloudellisuus voitti nopeuden – ja liiketaloudellisuus söi jalkatilan. Imperial Airways, never forget.

Sekä joukkoliikenne että yksityisautoilu ovat vuonna 2000 todella tehokkaita. Junat voivat kiittää turvallisesti maanpäällisissä tai -alaisissa putkissa, mikä nostaa liikenteen nopeutta ja vähentää riskejä. Ydinkeskustan liikenne voidaan siirtää magneettiraiteilla liikkuville auto-maattiautoille, jolloin ruuhkat vähenevät ja matkustajamäärät kasvavat.

Vaikka lentävistä autoista on haaveiltu yli sadan vuoden ajan, eivät ne ole yleistyneet. Myös perinteisemmät autot oli muotoiltu virtaviivaisesti ja futuristi-

sesti. Autoiluun liittyvät ennustukset ovat olleet enimmäkseen harhalaukauksia.

Atomivoiman piti olla ratkaisu kaikkeen – myös lentoliikenteeseen. Mannervälinen lentoliikenne olisi ydinkäyttöisillä matkustajakoneilla erittäin nopeaa, mutta todellisuus kamppasi nämäkin haaveet huolimatta siitä, että kylmän sodan aikana sekä Yhdysvalloissa että Neuvostoliitossa ydinkäyttöiset pommikoneet suorittivat koelentoja.

Autogiroit, pystysuoraan nousevat autot ja jopa lentävät lautaset nähtiin tulevaisuuden liikennemuotoina, sillä ne eivät tarvitse suuria lentoasemia eivätkä kiitoratoja. Lentävien lautasten ja atomilentobussien pysäkkejä ja muiden VTOL-koneiden laskeutumisalustoja suunniteltiin kaupunkirakennusten katoille. 🚀

### Tämän tahtoisin toteutuvan:

”4.10.2057. Sputnik 1:n satavuotispäivä. Avaruusajan alkua juhliitaan paitsi maassa, myös Kuussa, Marsissa, Europalla, Ganymedellä ja Titanissa – sekä Venuksen, Neptunuksen ja Pluton kiertoradoilla.”  
– Arthur C. Clarke, 1999

## MIKROKIVIKÄSI



UUSI, KAUNIS ALKU!

# Let the games begin! - MSI Gaming



## 138,90 €

### MSI Z87-G45 GAMING, Intel Z87, ATX-emolevy

MSI emolevyt on suunniteltu tarjoamaan pelaajille luokkansa parhaat ominaisuudet ja tekniikan. MSI:n Dragon tukemana, jokainen emolevy on tekniikan mestariteos joka on räätälöity pelaamisen täydellisyyteen.

Ominaisuudet:

LGA1150 CPU-kanta | Intel Z87-piirisarja | 3 x PCI-E 3.0 x16  
AMD CrossFire / NVIDIA SLI-tuki | 4 x DIMM DDR3-3000  
(OC) 6 x SATA III, 1 x mSATA | 6 x USB 3.0, 8 x USB 2.0 |  
Killer E2205 Gigabit Ethernet LAN



## 243,90 €

### MSI NVIDIA GeForce GTX 760 Twin Frozr 2GB

MSI:n näytönohjaimessa on valmiiksi tehtaalla ylikellotettu ydin, jota jäähdyttää erityisesti ylikellotettuja malleja varten suunniteltu Twin Frozr Gaming cooleri.

Ominaisuudet:

PCI-E Gen 3.0 | GPU kellotaajuus 1085/1150MHz (OC) |  
2GB GDDR5 6008MHz -muistia | DisplayPort, HDMI,  
1 x DVI-D, 1 x DVI-I

### Jimm's Punk Gamer

Jimm's Gamer -sarjan koneet ovat olleet jo pitkään pelaajien suosiossa ja suunnannäyttäjinä. Gamer -koneet ovat niittäneet mainetta arvosteluissa ja ovat moninkertaisia lehdistön -testivoittajia!

Punk Gamerista poistettiin vanhanaikainen savikiekko ja järjestelmälevyksi asennettiin nopea SSD-asema!

Ominaisuudet:

Intel i5-4670K 3.4GHz, 6Mt -prosessori | Intel Z87-piirisarjan emolevy |  
AMD Radeon R9 270X -näytönohjain | 120Gt, 2.5", Sata III -SSD asema |  
8Gt (2x4Gt) DDR3 1600MHz, CL9 -keskusmuisti | Cooler Master Silencio 550  
**Jimm's Edition** -kotelo | Windows 7 Home Premium 64-bit (Suomi)

## 1059 €

